

# abc

N° 80

COURS  
D'INFORMATIQUE  
PRATIQUE  
ET FAMILIALE

# INFORMATIQUE



Le Sanyo MBC-550 à l'essai

Construisons un robot

Les dictionnaires informatiques

Jeux de caractères en PASCAL

EDITIONS  
**ATLAS**

**Page manquante  
(publicité et colophon)**



# Des faits

**L'un des grands avantages de l'emploi des ordinateurs en pédagogie est qu'ils permettent d'accéder à de très grosses bases de données constamment remises à jour.**

Une « bibliothèque électronique » a bien des qualités; l'accès est immédiat, et il n'est pas besoin de quitter la salle de classe. Les éléments du savoir sont là en permanence, avec des informations remises à jour sans difficultés. Ce n'est évidemment pas le cas pour les livres où l'information contenue dans un texte est particulièrement statique et ne peut être changée que par la réimpression du livre lui-même. Mais le coût moyen de la fabrication d'un livre constitue un sérieux problème et tient déjà une très grande place dans le montant global des fournitures nécessaires à l'éducation. Les écoles sont d'ailleurs souvent obligées de garder plus longtemps qu'il en est besoin des éditions qui sont loin d'être à jour. A ces difficultés, il convient également d'ajou-

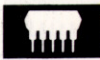
ter le prix du mot imprimé; il est déjà assez élevé pour gêner considérablement le développement des nouvelles méthodes d'enseignement à l'école. Une classe équipée d'ordinateurs n'aurait besoin, au pire, que de changer toutes ses disquettes, alors que remplacer toute une bibliothèque...

La consultation des bases de données, par le biais d'une liaison entre micro-ordinateurs et système de télécommunications, se révèle particulièrement fructueuse. Si l'information que vous recherchez est susceptible d'être retrouvée par la simple lecture d'une table des matières disposée en ordre alphabétique, dans de telles conditions, le livre est parfaitement adapté au problème. Mais si vous désirez connaître les noms de tous les animaux appartenant à la classe des chiroptères ou

## Ce monde qui nous entoure

On a parfois exprimé la crainte que l'emploi des ordinateurs dans les écoles n'amène inmanquablement les enfants à demeurer enchaînés à leur écran. Mais l'expérience a démontré que, en fait, cela conduisait à d'autres activités. Rassembler des informations est un processus multiforme, et il est parfaitement concevable que les élèves fassent là aussi des promenades dans la nature. (Cl. Steve Cross.)





encore tous les pays du globe où la part du produit national brut (P.N.B.) est inférieure à un certain pourcentage, la consultation des livres risque de vous devenir très vite insupportable. Il serait donc très utile de disposer, par exemple, de l'Encyclopédie AZ sur disquette.

Mais l'Encyclopédie AZ publiée par les éditions Atlas fait pas moins de quinze volumes (sans compter les suppléments annuels), et l'enregistrement de son contenu, ainsi que la mise en œuvre de facilités de recherches exigeraient une puissance de traitement énorme. C'est là que les grandes bases de données se révèlent indispensables. On se raccorde à elles grâce à un modem à 300 ou 1 200 bauds, et l'on a aussitôt accès à une quantité énorme de renseignements. Il existe ainsi des bases de données qui, par exemple, couvrent tout le domaine de l'électricité et de l'électronique et comportent plus de deux millions d'entrées ou des listes très complètes d'articles et autres matériaux de référence dans le domaine de la chimie. On ne pourra jamais, bien entendu, se passer de la lecture de ces articles; mais il faut d'abord savoir où et comment on peut les consulter.

Aux États-Unis, Dialog est l'une des plus importantes bases de données ouvertes au public, et les établissements d'enseignement recourent fréquemment à ses services. Dialog comprend deux cents bases de données différentes, contenant plus cent millions d'articles d'information: on peut y trouver des résumés portant sur tous les sujets, de l'art esquimau à la zoologie. L'utilisateur se voit préciser le titre et l'auteur de chaque article qui l'intéresse; ces articles sont condensés de façon très précise, à tel point que l'information recherchée a toutes les chances de s'y trouver. Tout cela peut être envoyé directement à votre imprimante, ou bien l'intégralité du texte vous parviendra, pour un prix tout à fait raisonnable. Il y a de plus en plus de bases de données de ce genre, et de plus en plus de logiciels spécialisés, dont la fonction est d'aider l'utilisateur à extraire et à trier l'information.

En Grande-Bretagne, Prestel a choisi une approche différente avec le système School Links: il propose des renseignements sur l'usage de l'informatique dans les écoles et crée une sorte de forum où l'on peut échanger des idées et nouer des contacts. Il est davantage destiné aux enseignants qu'aux élèves.

Les bases de données, cependant, ne servent pas seulement de « bibliothèque électronique »; elles sont appelées à jouer un rôle plus important — et plus actif — dans les salles de classe. Il existe déjà des programmes qui permettent aux enfants de traiter l'information et de la structurer de façon plus appropriée à leurs besoins; ils peuvent ainsi la rassembler, la trier et la mettre en forme comme ils l'entendent.

## Le projet Factfile

Toujours en Angleterre, une conférence tenue à l'université de Cambridge en 1981 débattit de l'emploi pédagogique des bases de données. Le débat inspira la création d'un programme nommé

## Données Dino

| NOM DE FICHIER<br>DINO | SUJET<br>DINOSAUR | JUSQU'À 10 RUBRIQUES          |                             |                    |
|------------------------|-------------------|-------------------------------|-----------------------------|--------------------|
|                        |                   | PREMIÈRE RUBRIQUE<br>LONGUEUR | SECONDE RUBRIQUE<br>ALIMENT | TROISIÈME RUBRIQUE |
| 1 FABROSAURUS          |                   | 1                             | VEGETAUX                    | TERRE              |
| 2 COELOPHYSIS          |                   | 2                             | VIANDE                      | TERRE              |
| 3 PLATEOSAURUS         |                   | 6                             | VEGETAUX                    | TERRE              |
| 4 SCOLOSAURUS          |                   | 6                             | VEGETAUX                    | TERRE              |
| 5 GUANADON             |                   | 5                             | VIANDE                      | TERRE              |
| 6 TYRANNOSAURUS        |                   | 15                            | VEGETAUX                    | TERRE              |
| 7 POLACANTHUS          |                   | 5                             | VEGETAUX                    | TERRE              |
| 8 HYPSILOPHODON        |                   | 6                             | VIANDE                      | TERRE              |
| 9 DEINONYCHUS          |                   | 2                             | VEGETAUX                    | TERRE, EAU         |
| 10 EUOPLIOCEPHALUS     |                   | 5                             | VEGETAUX                    | TERRE              |
| 11 BRACHIOSAURUS       |                   | 25                            | VEGETAUX                    | TERRE, EAU         |
| 12 STEGOSAURUS         |                   | 10                            | VEGETAUX                    | TERRE, EAU         |
| 13 APATOSAURUS         |                   | 18                            | VEGETAUX                    | TERRE, EAU         |
| 14 DIPLODOCUS          |                   | 28                            | VEGETAUX                    | TERRE, EAU         |
| 15 CORYTHOSAURUS       |                   | 9                             | VEGETAUX                    | TERRE              |
| 16 TRICERATOPS         |                   | 11                            | VIANDE                      | TERRE              |
| 17 ALLOSAURUS          |                   | 11                            | VIANDE                      | EAU                |
| 18 CERATOSAURUS        |                   | 7                             | POISSON                     | EAU                |
| 19 PLESIOSAUR          |                   | 12                            | POISSON                     | AIR                |
| 20 ICHTHYOSAUR         |                   | 12                            | POISSON                     | AIR                |
| 21 DIMORPHODON         |                   | 2                             | POISSON                     | AIR                |
| 22 PTERODACTYLUS       |                   | 8                             | INSECTE                     | AIR                |
| 23 PTERODACTYLUS       |                   | 2                             | INSECTE                     | AIR                |
| 24 RHAMPHORHYNCHUS     |                   | 2                             | VIANDE                      | AIR                |
| 25 QUETZALCOATLUS      |                   | 10                            |                             |                    |

### Une bonne mémoire

Factfile est une base de données très complète. Un fichier servant d'illustration, appelé Dino, est fourni avec le programme. Il contient des informations sur les dinosaures. Les enfants sont encouragés à se familiariser avec lui, et à en cataloguer le contenu (en haut), avant de créer leurs propres bases de données.

Vous voulez connaître tous les dinosaures avec

ALIMENT : VEGETAUX

HABITAT : EAU

Est-ce correct ?

Tapez OUI ou NON OUI

### Recherche de fichier

Recherche de fichier

Vous pouvez

A connaître tous les DINOSAURES

B connaître un DINOSAURE

C demander autre chose

D revenir à la base Choix

Tapez A, B, C ou D C

### Le menu de Dino

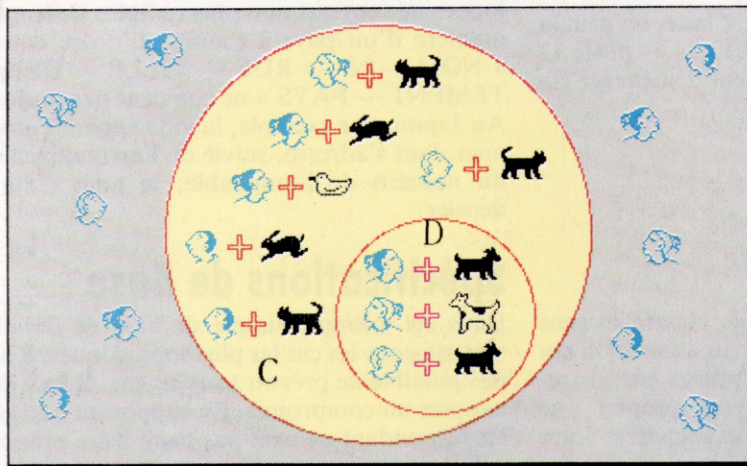
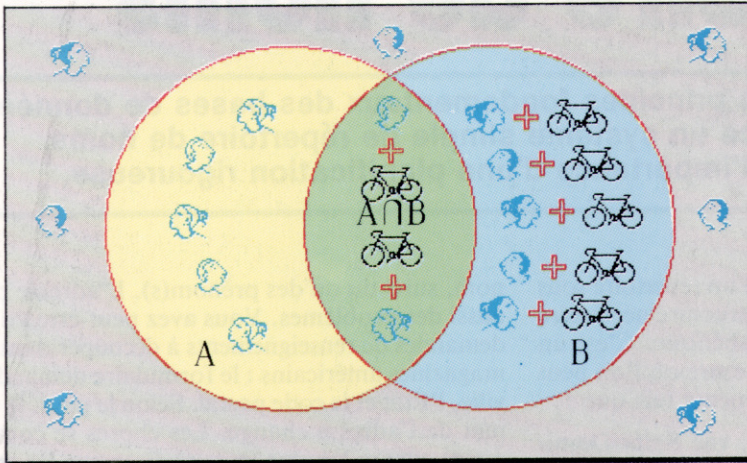
Factfile, dont le but est de permettre à de jeunes enfants de mettre sur pied des fichiers de données sur tout sujet qui peut les intéresser. Le rôle actif d'une base de données de ce genre dans la découverte et l'apprentissage apparaît très vite à l'observateur. Imaginons que les écoliers veuillent créer une base de données contenant des informations sur leurs professeurs. Il faudra d'abord qu'ils sachent quelles sont les catégories dont ils feront usage pour structurer leurs informations. Cela dépendra évidemment de ce qu'ils savent déjà, ainsi que de ce qu'ils peuvent découvrir eux-mêmes. Ils peuvent ainsi faire un choix selon divers critères comme l'âge, la taille, le sexe et le « type » de professeur. Il leur faut ensuite amasser les renseignements qui leur manquent. C'est alors que leur recherche prend une nouvelle direction; les données seront souvent des grandeurs physiques comme des mensurations... mais lesquelles? Le processus devient peu à peu plus complexe, à mesure que les enfants découvrent les implications des catégories qu'eux-mêmes ont mises au point. Que faut-il entendre par « type de professeur »? Est-ce une façon de les classer suivant la matière qu'ils enseignent ou de voir si tel ou tel est sympathique ou non? Ce dernier cas posera très vite des problèmes diplomatiques intéressants...

Une fois achevée la compilation des données, les élèves se rendent compte qu'il est possible de les corréler entre elles: ils voudront peut-être avoir la liste de tous les professeurs de mathématiques de sexe masculin, ou des enseignantes de français classées comme « gentilles ». On va encore plus loin lorsqu'on passe à l'analyse des résultats. Factfile permet leur stockage sur cassette ou sur disquette, et la sauvegarde d'informations ultérieures; les écoliers peuvent alors consulter leurs fichiers chaque fois que c'est nécessaire.

L'une des critiques les plus fréquentes consiste à dire que les enfants se retrouvent ainsi en permanence devant un écran, et se coupent peu à peu du monde extérieur qui les environne. Mais



## Travail d'ensembles



l'argument porte à faux. Outre qu'ils acquièrent une expérience directe des bases de données, ils sont peu à peu entraînés par *Factfile* à se livrer à toutes sortes d'activités sans rapport aucun avec un clavier de terminal. Ainsi, la création d'un fichier sur la faune locale les amènera à des recherches minutieuses, puis à créer leur propre système de classification.

*Your Facts* est un autre programme très populaire. Il est destiné à des enfants plus jeunes : on leur demande leur nom, leur sexe, on cherche à savoir s'ils ont un animal favori, une montre, une bicyclette, un frère ou une sœur. Lorsque tous ces renseignements ont été entrés au clavier, le programme demande *QUELQU'UN DESIRE-T-IL FAIRE UNE ENTRÉE SUPPLÉMENTAIRE?* Si la réponse est négative, on en revient au menu, et les enfants peuvent alors voir toutes les données, déterminer qui appartient à telle ou telle catégorie — ceux qui ont une montre, ou une bicyclette — ou peuvent jouer à un jeu. L'ordinateur doit trouver le nom d'un enfant en posant des questions du genre *ES-TU UN GARÇON?* ou *AS-TU UNE MONTRE?*, et ainsi de suite. Il fera enfin des propositions du genre *EST-CE QUE TU T'APPELLES NICOLAS?* *Your Facts* est une introduction parfaite à la notion même de base de données ; il encourage la lecture et l'écriture, met les enfants en contact avec le monde extérieur et, surtout, il constitue un exercice tout à fait passionnant.

### La classification de l'information

Interroger ou mettre sur pied une base de données est une activité qui a bien des correspondances avec d'autres aspects du travail scolaire. Les mathématiques modernes font ainsi appel très tôt à la notion d'ensembles. La théorie des ensembles est d'ailleurs largement consacrée au classement des données et à l'analyse des corrélations entre elles. C'est pourquoi la création d'une base de données est en fait un exercice assez familier qui pose des problèmes semblables. C'est ainsi que les concepts d'intersection et de sous-ensemble revêtent une importance particulière. L'intersection par exemple correspond à ce qu'on pourrait appeler un « chevauchement » entre deux ensembles, comme dans l'exemple du haut.

(Cl. Ian McKinnell.)

## One World

Une firme australienne, *Active Learning Software*, a lancé sur le marché une base de données très complète qui contient des informations sur pratiquement tous les pays du monde. Le programme s'appelle *One World*, et comprend des notes destinées à l'enseignant, ainsi que des feuilles à distribuer aux élèves. Chaque pays a droit à plus de trente types de renseignements différents : système politique, importations, exportations, langue, pays voisins, religion, taux d'alphabétisation, traités signés, pourcentage de population rurale et citadine, répartition de la population active (secteurs primaire, secondaire et tertiaire), pourcentage de sol cultivé, de forêts, de terrains désertiques... Le plus intéressant reste sans doute la facilité avec laquelle on peut rechercher et analyser les données — mais il est vrai que c'est un trait propre à toutes les bases de données.

Le menu principal propose différentes options :

1. AFFICHAGE D'UN PAYS DONNÉ
2. RECHERCHE A PARTIR D'UN SEUL CRITÈRE
3. ANALYSE A L'AIDE DE PLUSIEURS CRITÈRES
4. FONCTION AIDE
5. FIN DE LA SESSION

Supposons que l'un des élèves ait entendu parler d'un Monsieur Moubarak, qui est à la tête d'un pays du Moyen-Orient, et qu'il veuille savoir lequel. L'option 2 lui permettrait de taper son nom sous les rubriques 3 *CHEF DE GOUVERNEMENT* et 4 *CHEF D'ÉTAT*. Il ne faudrait pas longtemps au programme pour lui apprendre que *MOUBARAK* est à la tête de la république d'ÉGYPTE.

L'option *ANALYSE* est sans doute le point fort de ce logiciel. Les enfants peuvent analyser les informations, en utilisant jusqu'à trois options choisies parmi vingt critères différents. Pour trouver quel est le pays d'Europe dans lequel le pourcentage de population rurale est le plus élevé, ils n'auraient besoin que de deux renseignements — région et pourcentage de population. S'ils apprennent que les mineurs de bauxite se sont mis en grève au Surinam, ils pourraient très vite savoir où se trouve ce pays, apprendre que la bauxite est l'une de ses principales exportations, et que le conflit risque donc d'avoir de très sérieuses répercussions sur l'économie.

*One World* enseigne donc à rechercher et à interpréter les informations ; c'est un programme qui encourage les élèves à tirer leurs propres conclusions à partir des données qu'ils ont rassemblées. Il peut jouer un grand rôle dans les cours d'histoire, de géographie ou de sociologie.

Les bases de données ont une double fonction : elles donnent aux enfants des éléments qu'ils pourront retenir, et les encouragent à aller plus loin. De plus, ils peuvent observer comment l'ordinateur traite l'information, et voir quelle est la meilleure façon de formuler leurs questions. C'est là un apprentissage qui se révélera des plus utiles au lycée ou en faculté et, plus encore, dans la vie active.



# Répertoire de noms

**Après avoir donné les principes fondamentaux des bases de données, nous mettons en place un système simple de répertoire de noms, mettant en évidence l'importance d'une planification rigoureuse.**

Un des principaux avantages d'un répertoire sous forme de cahier est le peu de structure nécessaire à l'exception de l'ordre alphabétique. C'est un système très souple dans la mesure où l'on peut y faire figurer des enregistrements tels que :

PIERRE GROSJEAN, 16, rue Saint-Dominique 75006 (549.16.16.),

— Sa compagne s'appelle Claire, on peut la joindre au bureau au 660.12.14 — poste 13  
— appeler le soir — doivent déménager fin juin.

Suivi de :

GRUSE : 331.61.23.

Suivi de :

GRUV — voir « Bureau ».

Avec un répertoire, vous pouvez classer les gens selon leur prénom, leur nom, mais aussi selon des renvois avec clé indiquant d'autres enregistrements. Néanmoins, vous devrez adopter une approche plus systématique pour transférer votre fichier sur ordinateur.

Sachant qu'un répertoire informatisé est plus rigide que son équivalent manuel, essayons d'en concevoir un qui soit à même de répondre à toutes les situations. Nous aborderons chaque zone tour à tour, en commençant par le nom.

Les noms figurent en deux parties, d'abord le nom générique et, ensuite, le prénom ou partie spécifique de la zone Nom. Dans la plupart des pays, le ou les prénom(s) précèdent le nom. Dans d'autres, tels que la Hongrie, la Chine, le Japon, le nom précède le prénom. Au Japon, il n'y a que rarement un seul nom.

Vous voyez que les sujets prêtant à confusion ne manquent pas, et nous n'avons vu que la zone Nom ! Il faut donc définir des conventions précises sur l'emplacement du nom et des prénoms, et sur la longueur autorisée pour chacun. Les noms peuvent être aussi courts que Ng et plus longs que Cholmondley-Smythe. Il faut donc toujours prévoir les cas extrêmes en voyant « large ». Puisque le gestionnaire de base de données nécessite des longueurs de zone déterminées, nous devons donc fixer ces dernières en fonction des noms les plus longs que nous sommes susceptibles de rencontrer. Les inconvénients des longueurs de zones fixes seront abordés plus loin.

Dans la mesure où nous devons déterminer un format pour l'élaboration de la base de données, nous prendrons le nom comme zone clé principale. C'est pourquoi nous écrirons d'abord le

nom, suivi du ou des prénom(s). L'adresse pose aussi des problèmes. Vous avez peut-être vu des demandes de renseignements à découper dans des magazines américains : le formulaire demande la ville, l'État et le code postal. Selon le pays, le format de l'adresse change. Les choses se compliquent encore lorsque vous réalisez que les enveloppes ne sont pas non plus rédigées de la même manière d'un pays à l'autre. L'ordre courant « NOM — N° — RUE — VILLE — DÉPARTEMENT — PAYS » ne convient pas toujours. Au Japon, par exemple, la ville apparaît en premier dans l'adresse, suivie de l'arrondissement, du numéro de l'immeuble, le nom étant en dernier.

## Spécifications de base

Dans cet exemple simple de base de données, comme pour les cas les plus sophistiqués, il n'est pas possible de prévoir tous les cas. Il faut donc adopter un compromis. En supposant que votre correspondant ne peut pas avoir seize prénoms, voici les formats de zones pour un enregistrement :

- Zone Nom — jusqu'à 40 caractères.
- Zone Prénom — jusqu'à 60 caractères.
- Zone adresse : 1<sup>re</sup> ligne — jusqu'à 80 caractères.
- Zone adresse : 2<sup>e</sup> ligne — jusqu'à 80 caractères.
- Zone adresse : 3<sup>e</sup> ligne — jusqu'à 80 caractères.
- Zone adresse : 4<sup>e</sup> ligne — jusqu'à 80 caractères.
- Zone adresse : 5<sup>e</sup> ligne — jusqu'à 80 caractères.
- Zone Téléphone — jusqu'à 20 caractères.
- Zone Notes — jusqu'à 80 caractères.

Ce format de base devrait répondre à toutes les situations, même si des problèmes peuvent toujours survenir... Une difficulté potentielle vient de la longueur limitée de la zone Notes. Un autre problème peut venir du fait que le pays n'est pas précisé en tant que zone séparée. Selon la longueur de l'adresse, le pays se trouvera dans la troisième, quatrième ou cinquième zone, à moins qu'il ne figure nulle part. Cela ne devrait normalement pas être gênant, à moins de vouloir effectuer une recherche dans la base de données selon la clé du pays. Si votre base comprend de nombreux correspondants étrangers, incluez le pays dans une zone distincte. Ce genre de décisions doit toujours être prise lors de la conception de la base.

Le gestionnaire de base de données que vous utilisez sur votre ordinateur détermine les bases de données que vous pouvez mettre en œuvre.



Parmi les gestionnaires, Card Box, de Caxton, est un des systèmes les plus simples. Ses ressources pour ce qui concerne les recherches et le traitement des données sont limitées, mais il vous donnera des résultats satisfaisants dans la mesure où vos besoins seront correctement définis. Card Box ne constitue pas un langage avancé de programmation pour le maniement des zones et des enregistrements. Il vous permet néanmoins d'extraire des enregistrements spécifiques à partir de commandes simples au clavier. Si vous utilisez Card Box pour créer la base de données définie plus haut, vous devrez effectuer simplement une suite de saisies.

## Utilisation de Card Box

La première chose à faire est de donner un format aux enregistrements. Celui-ci détermine les informations qui y figureront l'index (quelles seront les zones clés), et la présentation des états à l'écran ou à l'imprimante. Si nous appelons le fichier de la base de données REPER, Card Box créera un fichier-format REPER.FMT. Ce dernier peut être édité afin de modifier si nécessaire la présentation des enregistrements. Il est également possible de créer d'autres fichiers-formats pour affirmer différemment les informations.

Comme la plupart des gestionnaires de bases de données, Card Box vous permet de saisir un texte permanent dans chaque zone. Pour un répertoire d'adresses, la signification de chaque zone est évidente : JEAN DURANT est bien le nom, pas une adresse; de même 666.33.55 est sans équivoque un numéro de téléphone. Mais avec d'autres bases de données, il vous sera utile de connaître l'intitulé des zones. D'où l'utilité de textes permanents. Comparez ces deux enregistrements :

06116  
3995  
86  
34,75  
DESCRIPTION Fuselage

et :

|                           |          |
|---------------------------|----------|
| NUMÉRO DE FABRIQUE        | 06116    |
| NOTRE NUMÉRO DE PIÈCE     | 3995     |
| NOMBRE DE PIÈCES EN STOCK | 86       |
| PRIX                      | 34,75 F  |
| DESCRIPTION               | Fuselage |

L'information est la même dans les deux cas, mais la deuxième formulation avec intitulés est plus lisible et élimine tout risque d'erreur.

Card Box autorise quatre attributs-index par zone : AUCUN, MAN(nuel), AUTO et TOUS. Dans une base de données répertoriant les articles en stock, PRIX ne sera pas une clé de recherche : on ne posera pas la question « Pièces dont le prix est inférieur à 500 F et supérieur à 400 F? ». En revanche, nous voudrions certainement savoir, par exemple « le nombre de pièces référencées #06116, en stock? ». C'est pourquoi nous ferons de NUMÉRO DE FABRIQUE une zone clé. Pour revenir

à notre exemple de répertoire d'adresses, nous utiliserons pour nos recherches le nom et probablement le prénom. Les deux seront des zones clés. Si le répertoire doit être utilisé commercialement, nous prendrons aussi comme clés la ville et l'indicatif téléphonique. Il est important de noter à cet égard que vous ne pouvez mettre au point une base de données efficace si vous ne savez pas très bien ce que vous en attendez. Définissez d'abord vos besoins. En effet, contrairement au vieux système du fichier dans la boîte à chaussures, une base de données se constitue en fonction de l'usage que l'on compte en faire.

### Création d'états

Megafinder vous permet de créer vos propres états. Celui ci-dessous est dérivé de l'état standard du programme, que l'on voit sur l'autre photo d'écran.

**Longueur de zone**  
Indiquée par des pointillés. La longueur totale de toutes les zones détermine le nombre d'états que peut contenir le fichier (nombre de fiches). Il est donc souhaitable que les zones soient les plus courtes possible.

**Intitulés de zones**  
Saisis par l'utilisateur lors de la création du format. Dans ce programme, ils ne servent qu'à l'utilisateur. Le programme se sert des titres pour afficher les informations. Il n'en tient pas compte lors du tri des données.

### Ligne de commande

Voici un aperçu des commandes que l'on a à sa disposition pour

l'édition d'un état. Il est possible d'insérer et de supprimer des lignes et des caractères.

**Identificateurs de zone**  
Ne pas confondre avec les intitulés. Les identificateurs donnent l'ordre d'accès des zones lors de la saisie. Les identificateurs permettent également le tri séparé ou groupé des zones. En utilisant l'option INDEX de Megafinder, et en allouant l'index 1 à la zone D, on obtient un tri rapide des enregistrements selon l'ordre alphabétique des noms de villes.

### Voici notre carte

Les systèmes de gestion de base de données peuvent présenter plusieurs méthodes de traitement de l'information. Cette photo d'écran de Megafinder, système de gestion de base de données (SGBD) destiné à l'Apple, montre un enregistrement en provenance du fichier « cartes de visite d'entreprises ».

Au-dessous, la liste des options de commande. Les données contenues par le programme peuvent être triées. Le SGBD peut également rechercher une donnée saisie parmi les données stockées (IDENTIFICATION), PASSER à des zones déterminées de la base, MODIFIER ou SUPPRIMER des informations, et AFFICHER ou IMPRIMER un état ou un rapport.



# Levons l'ancre

**Pour notre voyage vers le Nouveau Monde, nous pouvons lever l'ancre et mettre au point un module qui prendra en compte le trajet lui-même, et toutes ses obligations.**

Nous nous servirons d'une boucle FOR...NEXT très simple, et d'un compteur de boucle WK permettant de décompter les semaines de voyage. La limite supérieure de la boucle sera fixée par la valeur de la variable  $\mathcal{L}$  (fixée à huit en début de programme). De là, une série de sous-programmes gèrera les tâches qui doivent être actualisées toutes les semaines. La première d'entre elles consiste à tenir le journal de bord du capitaine, qui doit garder trace de l'état de santé de l'équipage, du niveau des provisions et ainsi de suite. Les lignes 800 à 889 s'en chargeront.

La force de chacun des membres de l'équipage dépend très largement de ce qu'il reçoit comme nourriture : pour rester en parfaite santé, il lui faut sa pleine ration de fruits, de légumes, de viande et d'eau. En début de voyage, la force de tous les matelots est fixée à 100 dans le tableau T\$(,). Le programme se livre à des calculs hebdomadaires pour s'assurer qu'il y aura assez de provisions de chaque catégorie pour toute la durée du périple. S'il vient à en manquer, le joueur se verra demander si l'équipage doit se contenter d'une demi-ration seulement. Cela ne sera possible que pour une semaine, et la force des hommes d'équipage diminuera de cinq points.

Il se peut aussi que telle ou telle provision soit complètement épuisée. Dans ce cas, vos marins perdront 10 points de force chaque semaine. Prenons un exemple plus complet : l'équipage manque de fruits et ne dispose que d'une demi-ration d'eau. Chacun de vos hommes verront leurs places hebdomadaires diminuées de 15 points. Si l'on vient à manquer totalement, ces mêmes forces sont réduites de 20 points par semaine. Si l'on peut reprendre pour la suite du voyage des rations normales, l'équipe voit ses forces remises automatiquement au point de départ la semaine qui suit.

Le sous-programme ligne 4000 donne, au début de chaque semaine, des renseignements sur la santé de chacun. Il distingue cinq catégories : en parfaite santé (de 75 à 100 points), en bonne santé (de 50 à 75), malade (25-50) et très malade (1-25). Tout marin dont la force tombe à zéro meurt ; mais le salaire du défunt devra cependant être payé à sa parenté lors de la rentrée au port, et reste à calculer chaque semaine. Ce sous-programme détermine aussi les salaires pour la semaine qui vient, les salaires déjà payés, et la somme d'argent qui reste. Le total des salaires est remis à zéro au début de cette section

(ligne 800), et se voit augmenté du total des gages de la semaine à chaque nouvelle traversée de la boucle. Si la somme est supérieure à celle qui vous reste (ce dont s'occupe le sous-programme ligne 5000), le joueur sera prévenu qu'il lui faudra commercer et faire des profits s'il veut être en mesure de payer les hommes.

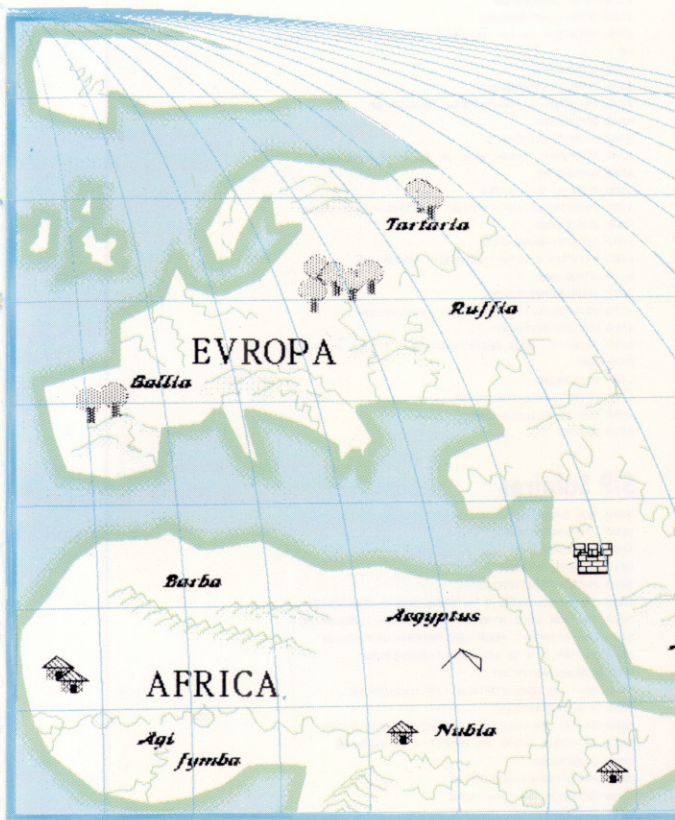
Si tout va bien, le voyage prendra huit semaines, mais des incidents imprévus peuvent toujours augmenter ce délai. La ligne 801 comporte une variable, H\$, qui indique si les membres de l'équipage survivent avec des demi-rations. La force de chacun (et sa catégorie) est établie par examen du tableau correspondant, T\$(,). Une boucle est mise en place ligne 4060 pour chaque matelot possible. T\$(, comprend des espaces blancs afin de pouvoir renfermer les informations relatives à un maximum de 16 matelots ; la boucle FOR...NEXT ira donc jusqu'à 16. Si une catégorie particulière est de zéro, la section correspondante du tableau sera vide ; la ligne 4070 dirige alors le programme vers la ligne 4110. En cas contraire, la ligne 4075 affiche la description.

## La mort d'un membre de l'équipage

Lorsque la force d'un matelot tombe à zéro, il meurt et son décès devra être consigné dans le journal de bord du capitaine. Tout décès fait naî-







la distribution des rations à l'équipage. Une boucle FOR...NEXT permet de traiter successivement chaque type de provision. Elle s'assure d'abord qu'il reste encore quelque chose à répartir. Il suffit pour cela de vérifier que la quantité de nourriture, contenue dans le tableau PA(I) est supérieure à zéro. Si ce n'est pas le cas, la force des marins est collectivement diminuée de 10 points. C'est une routine spécialisée (ligne 9300) qui s'en charge, en soustrayant de l'ancien chiffre la valeur de WF.

S'il reste encore de quoi se nourrir, l'ordinateur effectue une nouvelle vérification pour savoir s'il y en a assez pour le reste du trajet. La ligne 5180 effectue le calcul à l'aide de CN (nombre de membres de l'équipage), de PNI (quantité de nourriture pour une semaine) et du nombre de semaines qui restent avant l'arrivée aux Amériques. Si ce qui reste est inférieur aux premières prévisions, le joueur aura la possibilité de donner à ses hommes des demi-rations. Ce qui subsiste (rations pleines ou réduites de moitié) est indiqué par le tableau HR(I), Dimensionné ligne 802. Lorsque les provisions sont encore complètes, l'élément correspondant est fixé à 1; si une provision est réduite de moitié, à 0,5. Si l'équipage survit avec des demi-rations, le sous-programme de la ligne 9300 aura pour effet de diminuer de cinq points la force de chaque matelot.

Enfin, une ultime analyse voit si ce qui reste est inférieur à ce qui doit être distribué; si oui, vous ne pourrez partager que ce qu'il y a et, pour indiquer qu'il ne reste plus rien, l'élément correspondant du tableau sera fixé à - 999.

tre un problème : chaque section vide du tableau risque d'apparaître comme la représentation d'un mort. Pour résoudre la difficulté, le chiffre en question se voit affecter une valeur de - 999 pour une semaine, avant de repasser effectivement à zéro : la mort est alors affichée à l'écran en début de semaine suivante. Le programme cesse ensuite de prendre en compte le zéro, ce qui évite tout nouvel affichage des données relatives aux morts pendant le reste du voyage. Les lignes 4080 à 4098 évaluent la santé des autres hommes d'équipage; la ligne 4099 imprime les résultats. Le sous-programme prend fin à la ligne 4110, qui renvoie le programme à la recherche de la section suivante du tableau.

Une nouvelle boucle qui va de 4120 à 4135 permet de calculer le montant total des salaires; elle aussi est plusieurs fois parcourue, à raison d'une fois par catégorie. La ligne 4130 calcule les salaires relatifs à chacune d'elles : elle multiplie le nombre de marins appartenant à telle ou telle d'entre elles par le salaire hebdomadaire; le tout est stocké dans une variable, WW. Mise à zéro en début de chaque semaine par la ligne 4119, WW est affichée par la ligne 4145. Elle est alors ajoutée au total des sommes dépensées jusque là. La boucle comporte deux lignes, 4160 et 4165, qui provoquent l'impression des résultats à l'écran. Le joueur pourra toujours comparer la valeur obtenue à l'argent qui reste (ligne 4175). Le sous-programme qui commence à la ligne 5100 gère

## Variantes de basic

### Spectrum :

Effectuez les changements suivants :

```
847 LET I$=INKEY:IF I$=""THEN GOTO 847
4010 CLS
4190 LET I$=INKEY$:IF I$=""THEN GOTO 4190
4205 CLS
4295 LET I$=INKEY$:IF I$=""THEN GOTO 4295
4305 CLS
4398 LET I$=INKEY$:IF I$=""THEN GOTO 4398
5010 CLS
5080 LET I$=INKEY$:IF I$=""THEN GOTO 5080
5103 CLS
5220 INPUT: I$=LET I$=I$ (1 TO 1)
5298 LET I$=INKEY$:IF I$=""THEN GOTO 5298
```

### BBC Micro :

Effectuez les changements suivants :

```
4010 CLS
4190 I$=GET$
4205 CLS
4295 I$=GET$
4305 CLS
4398 I$=GET$
5010 CLS
5298 I$=GET$
```

## Module 4 : le voyage

### Longueur

```
40 JL=8:REM LONGUEUR VOYAGE
```

### Boucle principale voyage

```
800 WT=0
801 H$="N":REM INDICATEUR DEMI-RATION
802 DIMHR(4):HR(1)=1:HR(2)=1:HR(3)=1:HR(4)=1
820 FORK=1 TO JL:REM BOUCLE PRINCIPALE VOYAGE
825 GOSUB4000:REM SANTE EQUIPAGE
830 GOSUB4200:REM PROVISIONS
835 GOSUB4300:REM MARCHANDISES
840 GOSUB9200:PRINTCHR$(147)
842 PRINT:PRINT:PRINT
843 S$="IL EST PROBABLE QUE LE VOYAGE":GOSUB9100
844 PRINT "PRENDRA ENCORE":JL=WK+1:"SEMAINES"
845 GOSUB9200
846 PRINT:S$=K$:GOSUB9100
847 GETI$:IFI$="" THEN847
850 GOSUB5000:REM MONTANT SALAIRES
855 GOSUB5000:REM DISTRIBUTION RATIONS
889 NEXT WK
```

### Rapport santé équipage

```
4000 REM RAPPORT SANTE EQUIPAGE
4010 PRINTCHR$(147)
4020 S$=" JOURNAL DE BORD":GOSUB9100
4025 S$=" -----":GOSUB9100
4030 GOSUB9200
4035 PRINT "EN DEBUT DE SEMAINE":WK
4040 S$="L'ETAT DE SANTE DE L'EQUIPAGE EST":GOSUB9100
4045 GOSUB9200:PRINT
4055 PRINT
4060 FORT=1 TO16
4070 IFTS(T,1)=0 THEN4110
4075 PRINT$(TS(T,1))" (";
4078 IFTS(T,2)=999 THEN5$="MORT !!!!!!!":GOTO4099
9
4080 IFTS(T,2)75 THEN5$="EN PLEINE SANTE":GOTO4099
4085 IFTS(T,2)50 THEN5$="BIEN PORTANT":GOTO4099
4095 IFTS(T,2)25 THEN5$="MALADE !":GOTO4099
4098 S$="TRES MALADE !):"
4099 GOSUB9100:GOSUB9200
4110 NEXT T
4115 GOSUB9200:PRINT
```

### Salaires hebdomadaires

```
4119 WK=0
4120 FORT=1 TO5
4130 WK=WK+(CC(T)+HG(T))
4135 NEXT
4140 S$="SALAIRES HEBDOMADAIRES":GOSUB9100
4145 PRINTWK:"PIECES D'OR"
4150 GOSUB9200
4155 WT=WT+WK
4160 S$="SALAIRES VERSES JUSQU'ICI":GOSUB9100
4165 PRINTWT:"PIECES D'OR"
4170 GOSUB9200
4175 PRINT"IL RESTE":MO:"PIECES D'OR"
4180 PRINT:S$=K$:GOSUB9100
4190 GETI$:IFI$="" THEN4190
4199 RETURN
```

### Rapport provisions

```
4200 REM Rapport provisions
4205 PRINTCHR$(147)
4206 PRINT"EN DEBUT DE SEMAINE":WK:GOSUB9200
4210 S$="Il vous reste":GOSUB9100
4215 S$="CES PROVISIONS":GOSUB9100
4220 PRINT:GOSUB9200
4225 FORT=1 TO4
4226 IFPA(T)=0 ORPA(T)=999 THEN4240
4230 PRINTPA(T):US(T):"S DE":IP$(T)
4232 X=INT(PA(T)/CN*PN(T))
4235 PRINT "(ASSEZ POUR":X:"SEMAINES)"
4239 GOSUB9200
4240 NEXT
4290 NEXT
4295 PRINT:S$=K$:GOSUB9100
4295 GETI$:IFI$="" THEN4295
4299 RETURN
```

### Rapport marchandises

```
4300 REM "Rapport marchandises"
4305 PRINTCHR$(147)
4306 PRINT "EN DEBUT DE SEMAINE":WK:GOSUB9200
4310 S$="VOUS AVEZ AUSSI":GOSUB9100
4320 PRINT:GOSUB9200
4322 IFOR(1)=0 THEN4332
4325 PRINTOR(1):S$="FLACONS DE MEDICAMENT":GOSUB9100
4330 GOSUB9200
4332 IFOR(2)=0 THEN4342
4335 PRINTOR(2):S$="ARMES":GOSUB9100
4340 GOSUB9200
4342 IFOR(3)=0 THEN4352
4345 PRINTOR(3):S$="SACS DE SEL":GOSUB9100
4350 GOSUB9200
4352 IFOR(4)=0 THEN4362
4355 PRINTOR(4):S$="BALLON DE TISSU":GOSUB9100
4360 GOSUB9200
4362 IFOR(5)=0 THEN4372
4365 PRINTOR(5):S$="COUTEAU":GOSUB9100
4370 GOSUB9200
4372 IFOR(6)=0 THEN4380
4375 PRINTOR(6):S$="BIJOUX":GOSUB9100
4380 GOSUB9200:PRINT
4382 PRINT"IL VOUS RESTE":MO:S$="PIECES D'OR":GOSUB9100
4384 GOSUB9200
4397 PRINT:S$=K$:GOSUB9100
4398 GETI$:IFI$="" THEN4398
4399 RETURN
```

### S/P Salaires

```
5000 REM SALAIRES
5005 IFWT>MOTHEMS010
5008 GOTO5099
5010 PRINTCHR$(147)
5020 PRINT:PRINT:PRINT
5025 S$="L'EQUIPAGE A ENTENDU DIRE":GOSUB9100
5030 S$="QUE VOUS N'AVEZ PAS ABEZE":GOSUB9100
5035 S$="D'ARGENT POUR LES PAYER":GOSUB9100
5040 S$="EN FIN DE VOYAGE":GOSUB9100
5045 GOSUB9200:PRINT
5050 S$="ILS SONT FURIEUX":GOSUB9100
5055 GOSUB9200:PRINT
5060 S$="IL VA FALLOIR COMMERCER":GOSUB9100
5065 S$="POUR FAIRE DU PROFIT":GOSUB9100
5066 GOSUB9200
5070 PRINT:S$=K$:GOSUB9100
5080 GETI$:IFI$="" THEN5080
5089 RETURN
```

### S/P Rations

```
5100 REM RATIONS
5105 PRINTCHR$(147)
5105 S$=" DISTRIBUTION RATIONS":GOSUB9100
5106 S$=" -----":GOSUB9100
5107 GOSUB9200:PRINT "SEMAINE":WK:PRINT
5108 H$="N"
5110 FORT=1 TO4
5112 HR(T)=1
5115 IFPA(T)>0 THEN5180
5120 PRINT "non":P$(T):"? IL N'Y EN N'A PLUS!"
5130 S$="L'EQUIPAGE S'AFFAIBLIT":GOSUB9100
5135 WF=10:GOSUB9300
5139 GOTO5290
5180 X=(PN(T)+CN)+(JL-WK+1)
5185 IFPA(T)<X THEN5200
5190 GOTO5270
5200 PRINT "IL N'Y A PRESQUE PLUS DE":IP$(T)
5205 GOSUB9200
5210 S$="VOULEZ-VOUS DONNER A L'EQUIPAGE":GOSUB9100
5215 PRINT"DES DEMI-RATIONS DE":IP$(T)
5220 INPUTI$:I$=LEFT$(I$,1)
5221 IF I$<>"0" AND I$<>"N" THEN 5220:REM INPUT ERREUR
5225 IF I$="N" THEN 5270
5230 HR(T)=S:SIH$="0"
5240 WF=5:GOSUB9300
5250 S$="L'EQUIPAGE S'AFFAIBLIT":GOSUB9100
5270 X=PN(T)+HR(T)+CN
5272 IFX>PA(T) THENX=PA(T)
5275 PA(T)=PA(T)-X
5280 IFPA(T)=0 THENPA(T)=999
5285 PRINT:US(T):"S DE":P$(T):" DISTRIBUES"
5290 PRINT:GOSUB9200:NEXT
5295 PRINT:S$=K$:GOSUB9100
5298 GETI$:IFI$="" THEN5298
5299 RETURN
```

### Réduction force équipage

```
9300 REM REDUCTION FORCE EQUIPAGE
9310 FORS1=1 TO16
9320 TS(S1,2)=TS(S1,2)-WF
9330 IFTS(S1,2)<1 THENTS(S1,2)=999
9340 NEXT
9399 RETURN
```



# Un outil professionnel

**Le Sanyo MBC-550 représente une solution économique dans le marché des ordinateurs basés sur le microprocesseur 16 bits Intel 8088. Mais des insuffisances limitent cette machine.**

Malgré la baisse rapide des prix des processeurs 16 bits, la plupart des ordinateurs utilisant la technologie 16 bits sont presque toujours très chers. Cela peut s'expliquer par le type de clients qui s'intéressent à de tels ordinateurs. Ce sont, pour la plupart, des professionnels qui sont perçus par les fabricants comme des gens prêts à payer un peu plus. Cela est particulièrement vrai dans le cas des ordinateurs basés sur la série de processeurs Intel 8088, comme l'IBM.

Il est d'ailleurs intéressant de constater que, pour se « placer » sur ce marché où l'IBM PC est devenu un standard, la plupart des fabricants de micro-ordinateurs construits autour du processeur 8088 ont espéré qu'en proposant des produits moins chers que leur illustre concurrent, la clientèle se satisferait d'aussi bonnes affaires. Mais, encore récemment, ce secteur du marché était occupé par les ordinateurs de gestion construits autour du Z80, munis d'unités de disquettes et capables de « tourner » sous CP/M. De nombreux fabricants ont toutefois pris conscience du potentiel de ce secteur du marché professionnel et tentent de produire des ordinateurs 8088 peu coûteux.

## Incompatible?

Le Sanyo MBC-550 est l'une des premières machines à illustrer ce phénomène. Bien que Sanyo ne revendique pas la compatibilité avec l'IBM PC, le MBC-550 utilise en fait le système d'exploitation MS-DOS. La machine est disponible avec une ou deux unités de disquettes et est vendue environ 12 000 F.

Comme la plupart des ordinateurs de gestion, le MBC-550 est composé de deux unités distinctes, le clavier et l'ordinateur principal. La carcasse est en métal et en plastique de couleur gris métallique. Le clavier est divisé en trois sections. A l'extrême gauche, on retrouve cinq touches de fonction programmables qui, lorsqu'elles sont utilisées avec la touche « Shift », peuvent appeler jusqu'à dix fonctions qui varient selon l'application utilisée. En BASIC, ces touches peuvent servir à entrer des commandes comme LIST, LOAD et RUN.

Le clavier est bien disposé, toutes les touches de commande sont placées aux positions habituelles. Parmi les touches de commande, mentionnons un retour arrière effaceur et une touche « Insert/Delete », dont une seule fonctionne à l'intérieur de chaque application. Une touche « Graphics » est située à droite de la barre d'espa-

cement; cette touche appelle l'affichage de caractères graphiques lorsqu'elle est enfoncée. Un pavé numérique occupe l'extrême droite du clavier et peut servir de calculatrice ou de déplacement du curseur, si la touche « Number Lock » est appuyée.

L'ordinateur est assez large pour qu'on y pose le moniteur optionnel Sanyo. L'interrupteur d'alimentation est placé à l'avant de la machine. Il y a suffisamment d'espace pour deux unités de disquettes, bien que notre illustration présente le modèle à une seule unité. Les unités de disquettes utilisées par le MBC-550 sont du type standard 5 1/4 pouces. Devant chaque unité, une petite languette sert à maintenir la disquette en position et à interdire son retrait pendant une opération de lecture.

### Différences entre les deux versions

Le Sanyo MBC-550 est un ordinateur de gestion 16 bits bon marché qui fonctionne sous le système d'exploitation MS-DOS. La machine est proposée en deux versions; le MBC-550 est la version à unité de disquettes unique et la configuration à double unité de disquettes porte l'appellation MBC-555. (Cl. Chris Stevens.)



## Les interfaces

L'alimentation et les interfaces se trouvent à l'arrière de l'ordinateur. Du côté droit, on aperçoit le cordon d'alimentation, le porte-fusible et la connexion de prise de terre. A droite, l'interface parallèle Centronics sert à connecter une imprimante. A sa droite, une paire de prises de moniteur.



La première est une prise RVB, qui permet à l'ordinateur d'utiliser un moniteur couleurs, tandis que l'autre est une prise vidéo composite qui permet de brancher le moniteur monochrome vert Sanyo. Au-dessus de ces interfaces, on aperçoit d'autres ports d'extension qui sont destinés à prendre en charge ultérieurement des interfaces supplémentaires.

Le port Line, situé au-dessus de l'interface d'imprimante, sert à connecter une interface série RS232C. Cette interface permet à l'ordinateur de communiquer avec d'autres, par l'intermédiaire d'un modem, ou avec tout autre dispositif série, comme une imprimante.

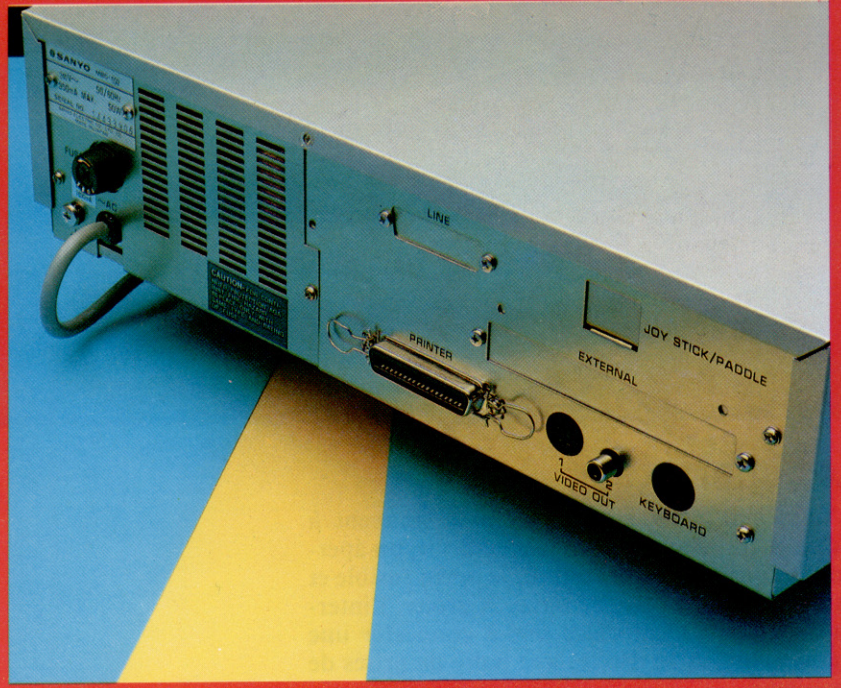
Tout près, on retrouve un port où il est possible d'installer une interface de la série MBC ou une interface de manche à balai compatible Apple. Il ne permet pas seulement d'adapter un manche à balai mais autorise aussi à commander l'ordinateur à partir de divers systèmes de type « souris ». Il existe encore une autre entrée pour un circuit extérieur permettant de connecter différents périphériques.

Contrairement à de nombreux fabricants qui recommandent fortement aux utilisateurs de ne pas toucher à l'intérieur de leurs machines, Sanyo donne dans le manuel utilisateur des instructions complètes pour monter ces interfaces, des modules de RAM et une seconde unité de disquettes. Cette initiative est très heureuse, mais le manuel mentionne que les utilisateurs doivent consulter un technicien compétent dans le moindre cas d'incertitude. Le reste du manuel est tout aussi pratique. Il renferme un glossaire de termes informatiques, une description complète du BASIC Sanyo — incluant des instructions relatives à l'utilisation des unités de disquettes à partir de BASIC — et de nombreux renseignements techniques.

Le BASIC Sanyo, une variante de la version Microsoft, est certainement adéquat, bien que ses

**Extension**

Bien que le MBC-550 ne soit pas doté de nombreuses options périphériques, Sanyo a conçu cette machine dans une perspective d'extension. On retrouve des fentes d'interface supplémentaires destinées à des manches à balai et à des périphériques série.



**Interface d'imprimante**

Ce port permet de connecter toute imprimante parallèle compatible Centronics.

**Puce d'interface de périphériques**

La puce 8255A commande les divers périphériques qui peuvent être connectés à l'ordinateur.

**Puce vidéo**

Cette puce permet d'utiliser toute section de la RAM de 256 K comme stockage de RAM vidéo.

**Haut-parleur**

Le potentiel sonore de l'ordinateur est exploité au moyen de ce haut-parleur intégré.



**Deuxième unité de disquettes**  
L'ordinateur dispose de l'espace permettant l'adjonction d'une deuxième unité de disquettes.

**Unité de disquettes**  
L'unité de disquettes unique équipant l'ordinateur est une unité de disquettes standard de 5 1/4 pouces.

**Fente ROM**  
Cet espace est réservé à l'addition d'un coprocesseur mathématique 8087.

**UC**  
Le MBC-550 utilise le microprocesseur Intel 8088 comme unité centrale.

**Puces RAM**  
Le MBC-550 standard est doté de 128 K de RAM. Notez cependant qu'il existe de nombreuses prises libres qui permettent à la machine d'accueillir des modules RAM additionnels.

**Alimentation**  
Le Sanyo MBC-550 possède son propre transformateur intégré situé à l'arrière du boîtier.

fonctions de traitements arithmétiques semblent être un peu lentes. La puce 8088 a depuis longtemps mauvaise réputation pour ce qui est du traitement mathématique (ce qui explique la popularité du coprocesseur mathématique 8087), mais le MBC semble remarquablement plus lent que d'autres machines construites autour du 8088. De même, le tracé de lignes est très peu pratique sur le MBC-550 car le BASIC ne comporte pas de commande DRAW; les lignes doivent donc être tracées au moyen de la commande PSET qui appelle simplement l'affichage d'un point à la couleur demandée. Le tracé d'une ligne implique donc l'utilisation d'une boucle.

## Baisse des prix

Outre les commandes Microsoft standard, comme MID\$, LIST et CIRCLE, il existe également des commandes servant à définir et à afficher des fenêtres. La présence de l'instruction conditionnelle WHILE...WEND encourage la programmation structurée.

De nombreux mots clés BASIC Sanyo peuvent être entrés en appuyant sur deux ou trois touches associées à la touche « Control ». Par exemple, le mot clé DIM peut être entré en appuyant simultanément sur les touches « CTRL », « Shift » et « D » tandis que PRINT est entré en pressant « CTRL » et « P ». Bien que toute approche visant à simplifier l'entrée des commandes BASIC doive être saluée et que de nombreuses séquences de touches aient un rapport avec les mots clés associés, il paraît improbable qu'une personne puisse se souvenir des quarante commandes disponibles en BASIC.

La disquette système MS-DOS est livrée avec le Sanyo MBC-550. Celle-ci renferme aussi le BASIC Sanyo, le populaire programme de traitement de texte WordStar et le tableur CalcStar. Ces programmes sont accompagnés d'un manuel qui décrit entièrement leur fonctionnement.

Il pourrait sembler plausible que le MBC-550 standard puisse sans grandes difficultés exécuter la vaste quantité de logiciels disponibles sur l'IBM PC. Mais il s'avère que, en raison de restrictions matérielles internes, aucun des programmes compatibles IBM que nous avons testés ne fonctionne sur cet ordinateur.

Il existe indubitablement un marché pour des ordinateurs de gestion MS-DOS à prix peu élevés et capables de « tourner » avec des logiciels compatibles IBM.

Le Sanyo MBC-550 représente la première tentative visant à se situer à un niveau raisonnable de prix, mais son absence de compatibilité maintient son isolement. Pour le responsable d'une petite entreprise, qui désire uniquement un ordinateur capable d'exécuter un programme de traitement de texte et un tableur efficaces, le MBC-550 peut constituer une solution intéressante. Cependant, toute personne qui désire avoir accès à une base logicielle plus étendue que celle offerte pour le Sanyo devra décider d'accroître légèrement son investissement, ou encore attendre quelque temps.

## SANYO MBC-550

### PRIX

MBC-550 standard : ★★★  
Version double MBC-555 : ★★★★★  
Les deux avec unité d'affichage.

### DIMENSIONS

375 × 355 × 108 mm.

### UC

Intel 8088 fonctionnant à 3,6 MHz.

### ÉCRAN

Écran texte de 80 × 25, donnant un mode haute résolution de 640 × 200 points. Jusqu'à huit couleurs sont disponibles en mode haute résolution.

### INTERFACES

Port parallèle Centronics avec options permettant d'installer une interface RS-232 ou un port de manche à balai compatible Apple.

### LANGAGE DISPONIBLE

BASIC Sanyo.

### CLAVIER

65 touches de type machine à écrire, 5 touches de fonction programmables doubles et un pavé numérique de 19 touches.

### DOCUMENTATION

Le manuel explique à fond les mots clés du BASIC mais ne propose malheureusement pas de travaux pratiques. Il présente une introduction à MS-DOS, à WordStar et à CalcStar. Fait inusité pour un ordinateur de gestion, le manuel de l'utilisateur donne des informations techniques très complètes, dont des instructions permettant aux utilisateurs d'installer leurs propres cartes d'extension.

### FORCES

Au prix proposé, le Sanyo MBC-550 semble certainement être un bon achat puisqu'il est ainsi possible d'acquérir un ordinateur de gestion 16 bits à un prix nettement inférieur au prix normal.

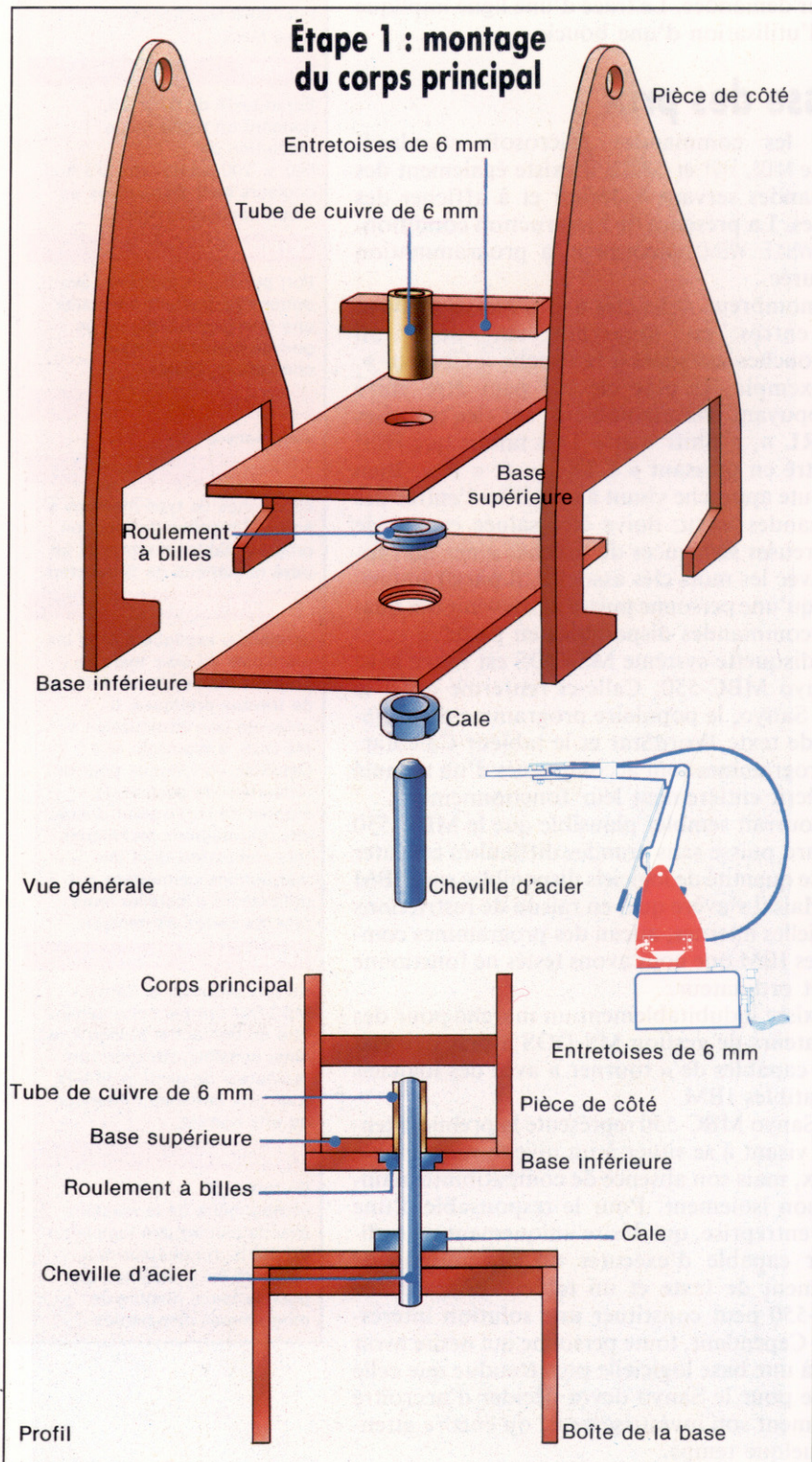
### FAIBLESSES

En raison de la non-compatibilité de la machine avec la plupart des logiciels existants, l'ordinateur doit résoudre les problèmes que pose le faible nombre de programmes disponibles.



# La construction du bras

Nous allons commencer la construction de notre robot en assemblant le corps principal, les sections inférieures et supérieures du bras, les joints des coudes et des épaules.



## Étape 1 : assemblage du corps principal

Percez un trou dans la partie inférieure de la base (la pièce la plus large), afin de monter le roulement à billes. Au moyen d'une mèche plus grosse ou d'une fraise, biseautez la lèvre supérieure du trou afin d'y poser le collet du roulement à billes. Puis percez un trou de 6 mm dans la partie supérieure de la base pour y introduire le tube de 25 cm de longueur et de 6 mm de diamètre interne. Après avoir aligné les trous et posé le roulement à billes sur la partie inférieure de la base, collez les deux pièces. Insérez vers le bas le tube de cuivre dans le trou et collez-le.

Lorsque cet assemblage est sec, les deux pièces latérales du corps principal peuvent être ajoutées. Percez les trous au haut des pièces latérales afin de pouvoir y introduire un tube de 4 mm de diamètre interne. Insérez un morceau de bois de 37 mm de longueur de  $6 \times 6$  mm comme entretoise entre les deux côtés — positionné à l'arrière de l'ensemble, en ligne avec la découpe du moteur de la pièce de côté — et collez le tout. Lorsque l'ensemble est sec, placez une cale (un gros écrou par exemple) sur la cheville d'acier en saillie au haut de la boîte de base que nous avons assemblée précédemment. Poussez la cheville dans le roulement à billes et vérifiez la libre rotation de l'ensemble du corps principal.

## Étape 2 : ensemble inférieur du bras

Coupez 37 mm de tube de 5 mm de diamètre extérieur et percez des trous au milieu des pièces inférieures du bras pour y insérer le tube. Veillez à ce que l'insertion ait un ajustage serré, puisque ce tube de cuivre fera partie du joint de l'épaule. Percez deux autres trous aux extrémités effilées des pièces inférieures du bras pour y introduire le tube de cuivre de 4 mm de diamètre externe. A partir du milieu, placez le tube de 5 mm dans les trous des pièces inférieures du bras en utilisant l'un des servomoteurs comme cale. Coupez deux morceaux de bois de 6 mm et de 20 mm de longueur comme entretoises et positionnez-les entre les deux pièces inférieures du bras. Ceux-ci se placent tout juste sous le tube de cuivre et servent à monter le servomoteur. Collez-les et, lorsqu'ils sont secs, montez le moteur en utilisant quatre vis à têtes et des rondelles de caoutchouc. L'ensemble inférieur du bras peut maintenant être assemblé au corps principal au niveau du joint de l'épaule. Insérez maintenant un tube de 4 mm et de 45 mm de longueur dans l'un des trous du haut du corps principal et dans le tube de cuivre déjà placé dans l'ensemble inférieur du bras.

Marquez la position correspondant à un alignement adéquat, démontez le tout et collez le gros tube de cuivre sur l'ensemble inférieur du bras. Assemblez de nouveau le joint de l'épaule lorsque c'est sec.

## Étape 3 : ensemble supérieur du bras

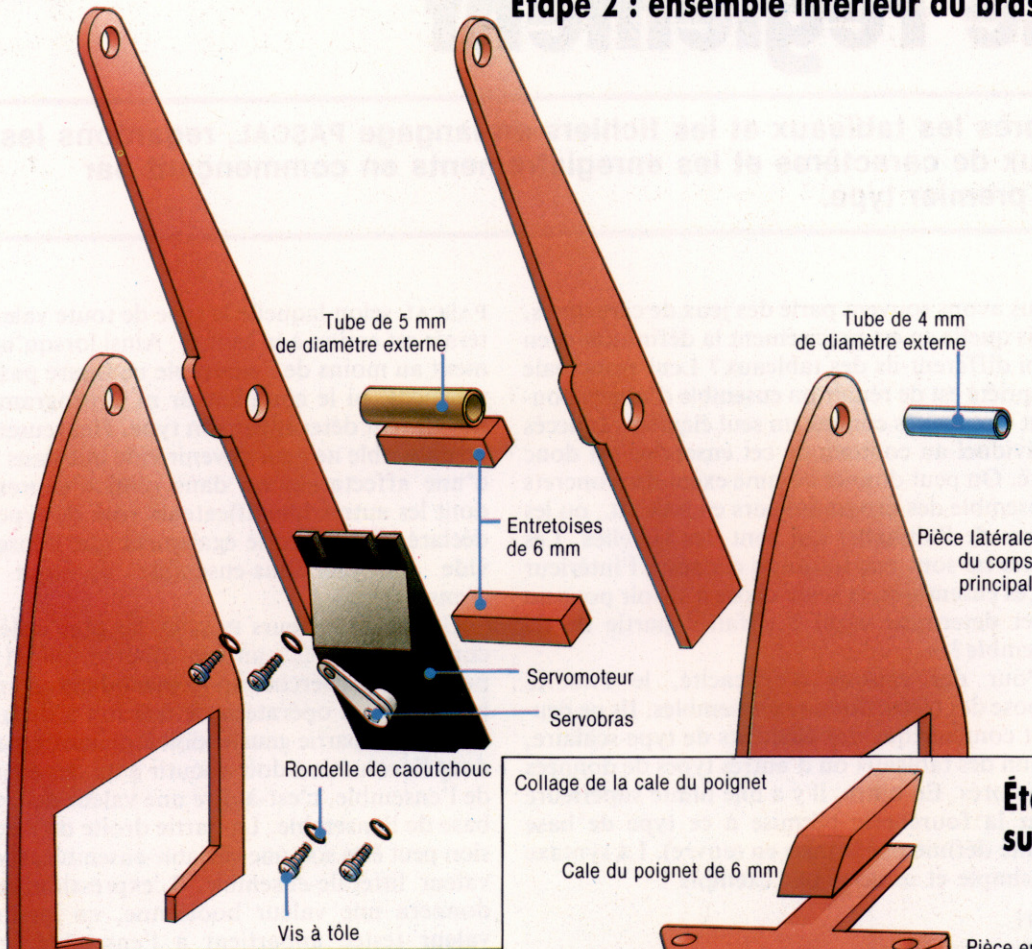
Percez deux trous pour insérer le tube de cuivre de 5 mm aux extrémités les plus larges des pièces supérieures du bras. Percez deux autres trous aux autres extrémités, assez gros pour introduire une vis mécanique. Collez un morceau de bois de 25 mm de longueur et de  $6 \times 6$  mm sur la pièce en « T » et arrondissez les extrémités. Cela servira de cale du poignet. Placez le « T » entre les deux pièces supérieures du bras et percez un trou à travers la cale du poignet afin de pouvoir insérer une vis à travers le poignet et à travers les pièces supérieures du bras. Fixez la pièce en « T » avec une rondelle et un écrou.

A l'autre extrémité du bras, placez un tube de 5 mm de diamètre et de 17 mm de longueur dans les trous percés, et assemblez l'ensemble supérieur du bras à l'ensemble inférieur au niveau du joint du coude en introduisant un tube de 4 mm de diamètre et de 25 mm de longueur.



Vue d'ensemble

## Étape 2 : ensemble inférieur du bras

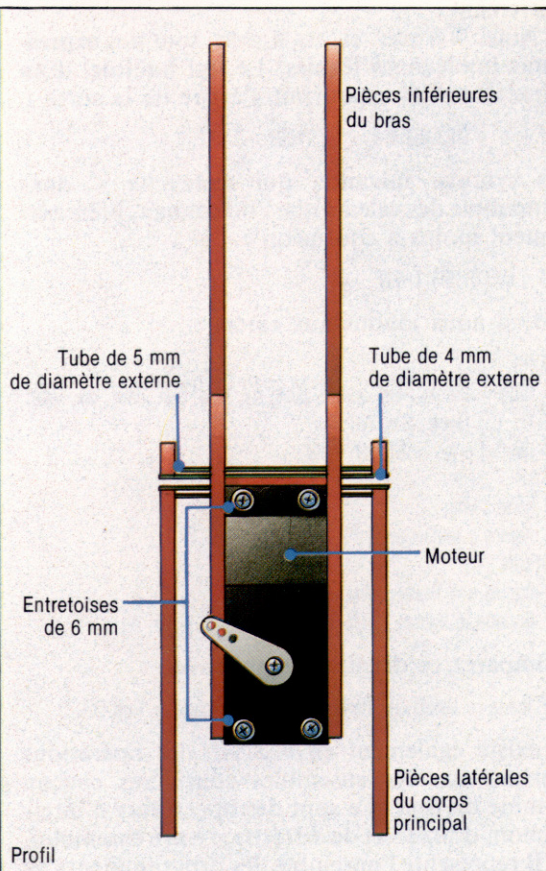
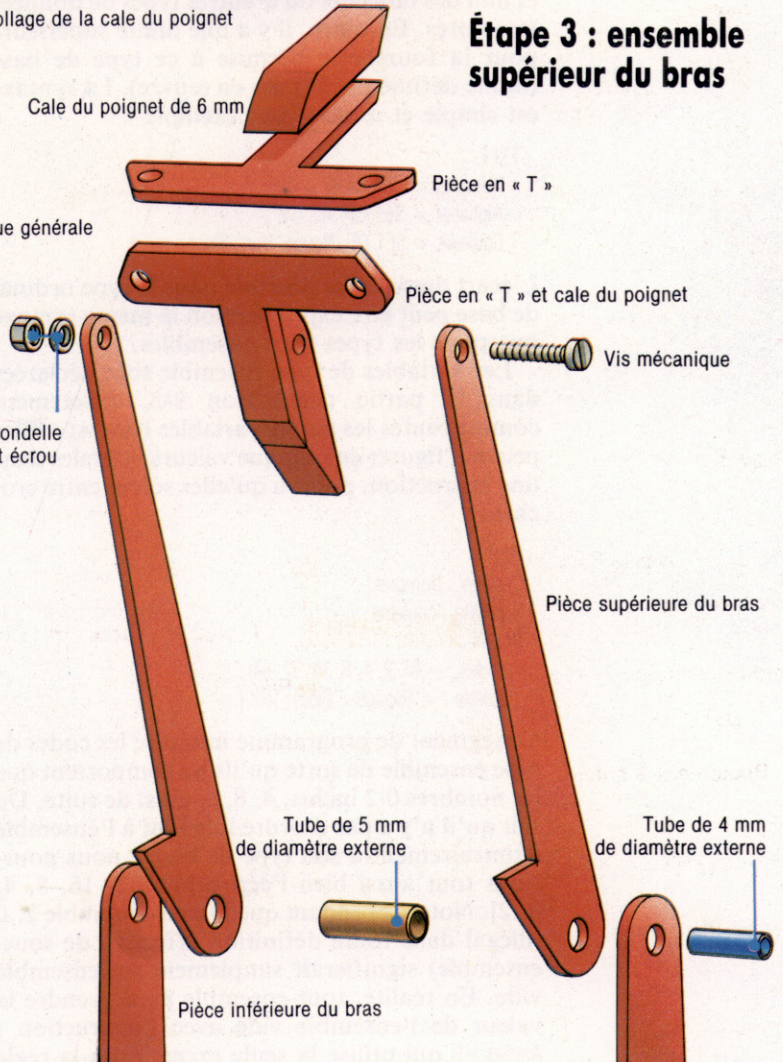


## Étape 3 : ensemble supérieur du bras

Collage de la cale du poignet

Cale du poignet de 6 mm

Vue générale



Profil

# Le règlement

**Après les tableaux et les fichiers en langage PASCAL, regardons les jeux de caractères et les enregistrements en commençant par le premier type.**

Nous avons souvent parlé des jeux de caractères, mais quelle en est précisément la définition et en quoi diffèrent-ils des tableaux? Leur principale propriété est de réunir un ensemble d'objets pouvant être traités comme un seul élément. L'accès individuel au contenu de cet ensemble est donc évité. On peut donner comme exemples concrets l'ensemble des programmeurs en PASCAL, ou les lettres de l'alphabet qui sont des voyelles. Les objets ne sont pas toujours classés à l'intérieur de ces ensembles, la seule chose à savoir pour un objet déterminé étant : « fait-il partie de tel ensemble? ».

Pour des raisons d'efficacité, le PASCAL impose des restrictions aux ensembles. Ils ne peuvent contenir que des éléments de type scalaire, et non des tableaux ou d'autres types de données structurés. En outre, il y a une limite supérieure pour la fourchette permise à ce type de base (limite définie par la mise en œuvre). La syntaxe est simple et directe, par exemple :

```
TYPE
  Nombres = SET OF 0 .. 127;
  Alphabet = SET OF 'A' .. 'Z';
  Couleurs = SET OF (Rouge, Vert, Bleu);
```

L'écart des valeurs possible pour le type ordinal de base peut être exprimé selon la même syntaxe que pour les types sous-ensembles.

Les variables de type ensemble sont déclarées dans la partie déclaration VAR, exactement comme toutes les autres variables PASCAL. Elles peuvent figurer en tant que valeurs littérales dans une instruction, pourvu qu'elles soient entre crochets :

```
VAR
  codes : Nombres;
  palette : couleurs;
BEGIN
  codes := [0..2, 4, 8, 16, 32, 64];
  palette := [Rouge .. Bleu]; {etc.}
```

Ce segment de programme initialise les codes de type ensemble de sorte qu'ils ne comportent que les nombres 0-2 inclus, 4, 8, et ainsi de suite. Du fait qu'il n'y a pas d'ordre inhérent à l'ensemble (contrairement à son type de base), nous pourrions tout aussi bien l'écrire [64, 32, 16, 8, 4, 0..2]. Notez cependant que le sous-ensemble 2..0 (illégal dans toute définition effective de sous-ensemble) signifierait simplement un ensemble vide. En réalité, tout ensemble peut prendre la valeur de l'ensemble vide avec l'instruction : AnySet := [] qui utilise la seule exception à la règle

PASCAL selon laquelle le type de toute valeur littérale est connu à la lecture. Ainsi lorsqu'un élément au moins de l'ensemble ne figure pas littéralement, ni le compilateur ni le programmeur ne peuvent déterminer son type. Heureusement, un ensemble ne peut devenir vide que dans le cas d'une affectation ou dans celui d'expressions dont les autres identificateurs sont de type déjà déclaré. Cela signifie également que l'ensemble vide est un sous-ensemble de tout type d'ensemble.

Un des opérateurs PASCAL les plus utiles est, comme DIV et MOD, un mot réservé : IN. Il nous permet de rechercher un élément dans un ensemble. C'est un opérateur de relation à deux opérandes. La partie gauche doit être une expression dont l'évaluation doit aboutir à un des éléments de l'ensemble, c'est-à-dire une valeur du type de base de l'ensemble. La partie droite de l'expression peut être soit une variable-ensemble, soit une valeur littérale-ensemble. L'expression entière donnera une valeur booléenne, vraie lorsque la valeur testée appartient à l'ensemble, et faux autrement.

Ainsi "N IN codes" et "Vert IN palette" sont des expressions booléennes légales. Le test habituel d'un caractère-chiffre pourrait s'écrire de la sorte :

```
IF (c >='0') AND (c <='9') THEN...
```

La syntaxe suivante, qui recherche 'c' dans l'ensemble des valeurs que l'on connaît, prête nettement moins à confusion :

```
IF c IN ('0'..'9') THEN...
```

Ou, si nous jouons aux cartes :

```
TYPE
  rang = (deux, trois, quatre, cinq, six, sept, huit, neuf, dix, Valet,
  Dame, Roi, As);
  JeuDeCartes = SET OF rang;
VAR
  carte : rang;
  figures : JeuDeCartes;
BEGIN
  figures := (Valet .. As);
  IF carte IN figures THEN {etc.}
```

Comparez ce dernier exemple avec :

```
IF (carte = Valet) OR (carte = Reine) OR (carte = Roi) OR ...
```

Il existe également en PASCAL des opérations portant sur des ensembles considérés chacun comme un tout. Ce sont des opérateurs d'intersection, d'union et de différence entre ensembles. Si B représente l'ensemble des programmeurs en



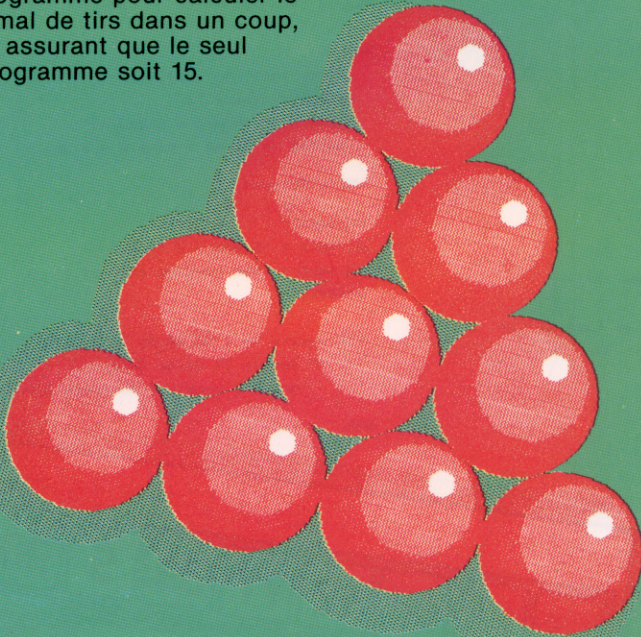
## Billard

Ce jeu de billard utilise quinze boules rouges à un point, une boule blanche « signal », et six boules de couleur rapportant des points selon le barème ci-dessous :

|        |   |
|--------|---|
| jaune  | 2 |
| vert   | 3 |
| marron | 4 |
| bleu   | 5 |
| rose   | 6 |
| noir   | 7 |

Après avoir placé une boule rouge, le joueur a droit à essayer une boule de couleur. Les boules de couleur placées sont retirées et remises sur la table. Après la dernière boule rouge placée, toutes les boules de couleur doivent être placées dans l'ordre croissant des couleurs.

Un « coup » à ce jeu de billard est une suite de boules placées, ou un mauvais tir, ou la fin du jeu (plus de boules). Écrivez un programme pour calculer le nombre maximal de tirs dans un coup, mais en vous assurant que le seul nombre du programme soit 15.



Ian McKinnell.

BASIC, et P, celui des programmeurs en PASCAL, l'intersection des deux ensembles désigne les programmeurs utilisant le BASIC *et* le PASCAL. L'union de B et de P est l'ensemble des programmeurs en BASIC ou en PASCAL.

$P - B$  représente les programmeurs programmant en PASCAL, mais non en BASIC. De manière similaire, la notation utilisée pour l'union est  $P + B$ , et  $P * B$  pour l'intersection. Ces opérateurs sont les mêmes que les opérateurs arithmétiques, mais les opérations sont très différentes et ne doivent pas être confondues. La raison pour laquelle ils interviennent dans le contexte des opérations sur les ensembles est leur utilisation dans le test de bit.

Prenez un ensemble de huit éléments. La présence d'un élément donné pourrait être indiquée par la représentation d'un certain bit selon une disposition à huit bits (un octet). L'absence de l'élément peut être indiquée de manière similaire par zéro. Les tests d'appartenance pourront donc simplement être effectués par un test de bit ou de masque. Une opération d'union entre ensembles devient une opération de type OR, l'intersection étant une simple opération AND. Ces opérations existent au niveau code machine. Le compilateur PASCAL peut donc fournir des syntaxes de données de type ensemble, ainsi que les opérations qui y ont trait. Le supplément de mémoire est minime, spécialement lorsque la taille des ensembles reste à la mesure de celle d'un mot pour l'ordinateur, par exemple sur des machines à 32 bits et davantage. C'est pourquoi les opérations portant sur des ensembles sont efficacement réalisées en PASCAL.

Nous pouvons maintenant répertorier de manière satisfaisante tous les opérateurs PASCAL. Ceux que nous n'avons pas traités explicitement sont communs à tous les autres langages. Le PASCAL simplifie les choses en n'acceptant que quatre degrés de préséance ou priorité. La priorité absolue revient aux opérateurs monadiques (à un seul opérande) : ce sont les signes + et - qui indiquent le signe d'un nombre et l'opérateur de négation booléen NOT. Le deuxième degré de priorité est affecté à tous les opérateurs de « multiplication ». Le troisième concerne l'addition et la soustraction ; le dernier est affecté aux opérateurs de relation dont IN.

Vous remarquez que les deux autres opérateurs booléens AND et OR sont considérés à juste titre comme des opérateurs de multiplication et d'addition (respectivement). Cela reflète fidèlement l'algèbre booléenne, et signifie que de nombreux tests de relation sont mis entre crochets pour supprimer ces ordres de priorité. Faute de quoi, par exemple, `IF N > 0 AND N < 10 THEN...` aboutira à une erreur à la compilation, parce que l'expression `0 AND N` (qui doit être évaluée en premier) essaie de combiner deux opérands d'entiers et un opérateur booléen.

C'est peut-être encore une mauvaise manie héritée du BASIC ? Voilà en tout cas l'occasion de vous en débarrasser... Naturellement, les opérateurs de même degré de priorité sont évalués de la gauche vers la droite, comme d'habitude.

## Priorité entre opérateurs

Le tableau suivant indique l'ordre des priorités entre opérateurs PASCAL. Les parenthèses assurent qu'une expression sera évaluée séparément, supprimant ainsi l'ordre normal :

| Priorité | Opérateurs       | Type             |
|----------|------------------|------------------|
| Maximale | NOT + -          | {monadique}      |
|          | AND * / DIV MOD  | {multiplication} |
|          | OR + -           | {addition}       |
| Minimale | << = <> > = > IN | {de relation}    |

## Bingo

Une carte de Bingo peut être considérée comme un ensemble. Bien que les éléments de base (des entiers compris entre 0 et 90) soient ordonnés, la seule question d'importance est de savoir si le numéro appelé appartient à une carte déterminée. En termes PASCAL : 'nombre IN carte' est soit vrai soit faux.

Le programme simule un jeu de Bingo en lisant des chiffres au clavier. Après chaque nouveau chiffre, l'ensemble des chiffres déjà appelés est testé par rapport à l'ensemble-carte.

L'expression Carte-Appelée se rapporte à l'ensemble vide lorsque tous les éléments de la carte figurent également

dans l'ensemble Chiffres-Appelés.

Ajouter un élément à un ensemble qui le contient déjà ne modifie pas cet ensemble, tout comme retirer un élément qui n'y figure pas (!). Remarquez que l'on ne peut ajouter directement un élément à un ensemble : il faut d'abord créer un ensemble à un élément en mettant l'élément entre crochets. Après, on effectue l'union des deux ensembles. Tel qu'il est, le programme permettrait des entrées en double dans une carte, et accepterait un nombre en dehors de l'écart autorisé 0-90, suscitant des erreurs à l'exécution. Comme exercice, essayez de lui ajouter des boucles simples qui rejettent les nombres déjà appelés et remédient aux deux anomalies citées plus haut.

```
PROGRAM Bingo (entrées,sorties)
```

```
CONST
```

```
  colonnes = 40; {adaptation vidéo}
  lignes = 25;  { }
```

```
TYPE
```

```
  BINGO = ENSEMBLE DE 1 .. 90;
```

```
VAR
```

```
  compte : 1 .. 15;
  Autorisé,
  Appelé,
  Vide,
  Carte : BINGO;
  nombre : entier;
  Maison : booléen;
```

```
BEGIN
```

```
  Vide := [];
  Autorisé := [1 .. 90];
  WriteLn ( '*** LOTO ***' : Lignes );
  WriteLn;
  WriteLn ( 'donnez les 15 numéros de cartes,' );
  WriteLn ( '(suivi chacun de RC) :' );
  Carte := Vide
```

```
FOR compte := 1 TO 15 DO
```

```
  BEGIN
```

```
    write ( compte : 10, ' ? ' );
    ReadLn ( numéro );
    Carte := Carte + [numéro]
  END;
```

```
  WriteLn;
```

```
  WriteLn ( ' REGARDEZ!' : lignes);
```

```
  WriteLn;
```

```
  WriteLn ( 'appelez chaque nombre : ' );
```

```
  Appelé := Vide;
```

```
  REPEAT
```

```
    write ( '?' : lignes );
    ReadLn ( nombre );
    Appelé := Appelé + [nombre];
    Maison := Carte - Appelé = Vide
```

```
  UNTIL Maison;
```

```
  WriteLn;
```

```
  WriteLn ( 'Bravo ! : lignes' );
```

```
  WriteLn ( 'Vos nombres étaient :' );
```

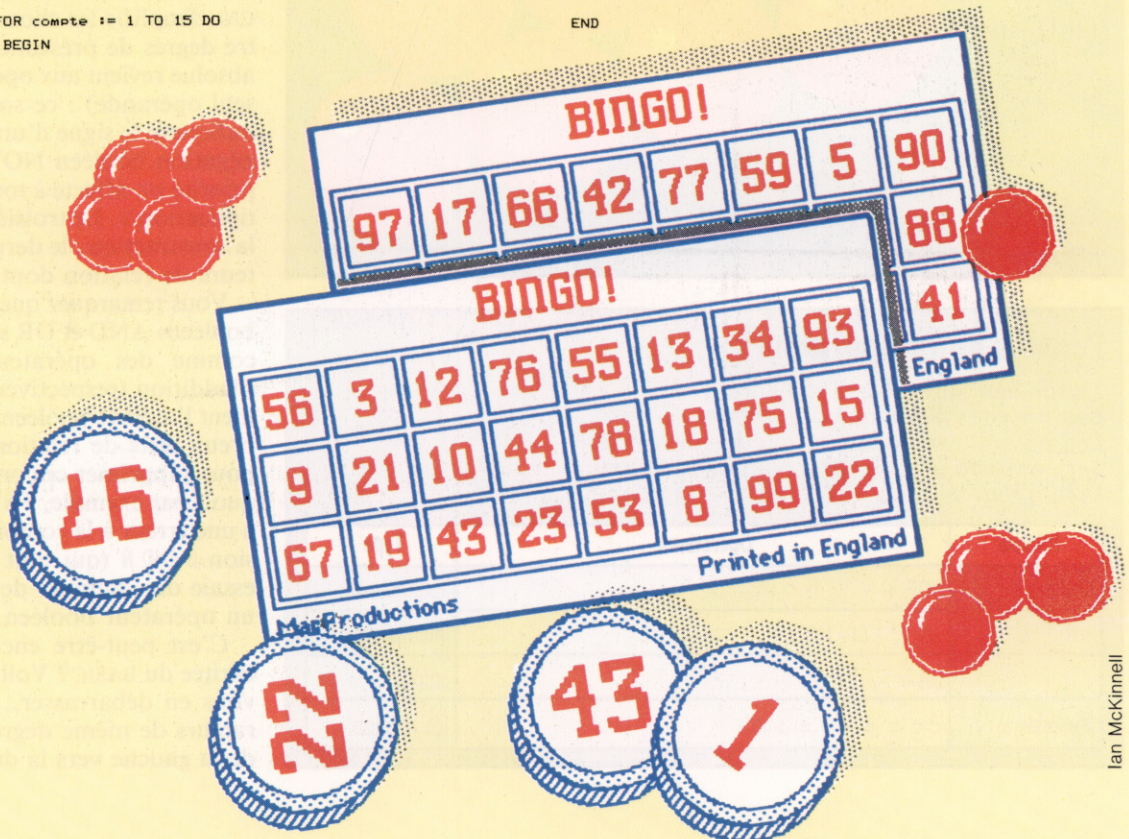
```
  WriteLn;
```

```
  FOR nombre := 1 TO 90 DO
```

```
    IF nombre IN Carte THEN
```

```
      write ( nombre : Colonnes DIV 8 )
```

```
  END
```





# Les dictionnaires

Bien des mots anglo-saxons « traînent » encore, même dans les textes français. Pour vous aider à débroussailler le terrain, voici une sélection de dictionnaires et lexiques.

Alors comment s'y retrouver dans la « jungle » des termes informatiques ? Chaque mois, de nouveaux mots surgissent, qui correspondent à des concepts sans équivalent dans le langage courant. Très souvent, ils sont anglais. Cela peut poser des problèmes.

Parmi les ouvrages présentés ici, vous trouverez indifféremment des dictionnaires français-anglais, ou vice versa, et des lexiques et glossaires. En effet, il serait arbitraire de faire une distinction entre ces deux catégories, étant donné que la plupart des lexiques expliquant les termes donnent aussi leur équivalent anglais, et que les dictionnaires donnent, avec la traduction des mots, leur explication informatique.

Pratiquement, tous les dictionnaires donnent les termes anglais. Pourquoi cette langue est-elle prééminente ? Sans doute parce que c'est dans les pays anglo-saxons (États-Unis et Grande-Bretagne) que le développement de l'informatique a été le plus rapide et, surtout, parce que c'est dans cette langue que se font la plupart des publications scientifiques ou techniques.

Mais il ne faut pas perdre de vue le fait que tous ces termes ont leur équivalent en français, et c'est aussi dans un souci de préserver l'intégrité de notre langue qu'il est important de se référer à ces ouvrages, afin de ne pas tomber dans un jargon franglais encore trop répandu parmi ceux qui pratiquent l'informatique.

## HARRAP'S DICTIONNAIRE INFORMATIQUE Anglais-Français Français-Anglais

Pour comprendre  
le « langage » international  
de l'informatique

### Harrap's Dictionnaire informatique anglais-français/français-anglais

Pour comprendre le « langage » international de l'informatique.

Face au développement considérable du langage technique se rapportant à l'ordinateur, et tout en tenant compte du nouveau vocabulaire américain, voici que la « bible des dictionnaires bilingues » se met à l'informatique, avec plus de 25 000 mots et locutions. Symbole des années quatre-vingts, synthèse des années soixante-dix, cet ouvrage est d'une grande actualité.

130 pages. Format 16 cm x 24 cm.  
Harrap. Diffusion Bordas.

### Dictionnaire d'informatique bureautique-télématique

Dictionnaire bilingue en deux volumes :

- Anglais-français, 276 pages.
- Français-anglais, 220 pages.

Ces deux volumes sont complétés et actualisés par de nouveaux termes et acceptions.

Par M. Ginguay.  
Masson.

### Lexique microprocesseurs Dictionnaire et définitions

Le but de ce petit lexique de poche est de fournir une référence commode à tous ceux qui travaillent dans le domaine des micro-ordinateurs ou qui s'y intéressent.

La première section est un dictionnaire anglais-français. La deuxième section donne les abréviations anglaises et françaises, et les définitions des termes français avec leur traduction en anglais. La troisième section donne la définition des signaux dans les trois standards principaux (RS232, IEEE 488, S100). La quatrième partie présente un guide qui permet de s'y retrouver dans la jungle des numéros de boîtiers utilisés dans les micro-ordinateurs. Enfin, la dernière partie fournit les adresses des fabricants les plus importants et de leurs représentants en France.

145 pages. Format 10,5 cm x 14 cm.  
Sybex.



### Les mots clés de l'informatique

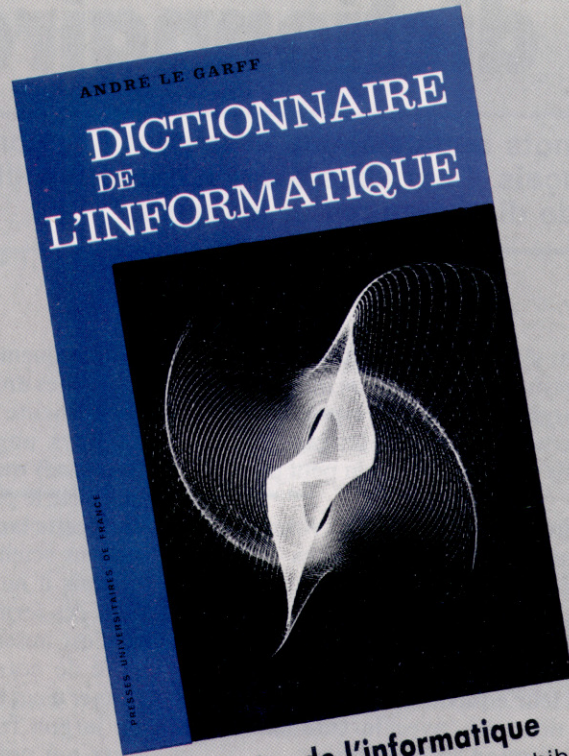
Ce lexique au format très pratique donne une définition simple, claire et complète des mots essentiels de l'informatique.

Il provoquera le désir d'approfondir ses connaissances techniques et permettra de mieux maîtriser le domaine du traitement de l'information, tout en le rendant accessible.

Débutant par la liste des cinq mots fondamentaux à découvrir en priorité, il présente ensuite par ordre alphabétique l'ensemble des mots clés essentiels.

Toutes les définitions sont suivies d'une explication plus technique et concrète, illustrée par un schéma ou un dessin en deux couleurs.

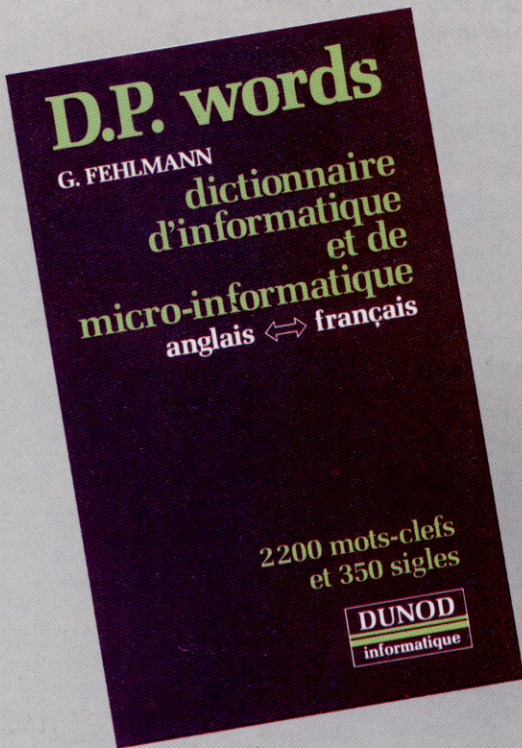
Éditions Foucher.



### Dictionnaire de l'informatique

Un ouvrage indispensable dans la bibliothèque de tout informaticien, relié pleine toile, titre or, sous jaquette illustrée en couleurs.

Par André Le Garff.  
584 pages. Format 16,5 cm x 25 cm.  
Presses Universitaires de France.



### D.P. Words Dictionnaire d'informatique anglais-français, français-anglais

Ce dictionnaire comprend 2 000 mots clés et 300 sigles, qui composent la terminologie essentielle, actuelle et vivante du traitement de l'information. Une explication sur la signification technique du mot est donnée, quand cela est nécessaire.

Par G. Fehlmann.  
176 pages.  
Dunod Informatique.

### Dictionnaire d'informatique

Ce dictionnaire français réunit, dans un contexte très sobre, tout le vocabulaire nécessaire : noms courants ou moins courants, noms de produits et de réseaux de constructeurs, sigles. Lexique spécialisé, il présente l'avantage d'être facile à consulter. Ses 1 700 entrées et sous-entrées sont complétées par plusieurs annexes : abréviations anglaises et françaises, codes, normes nationales et internationales, etc.

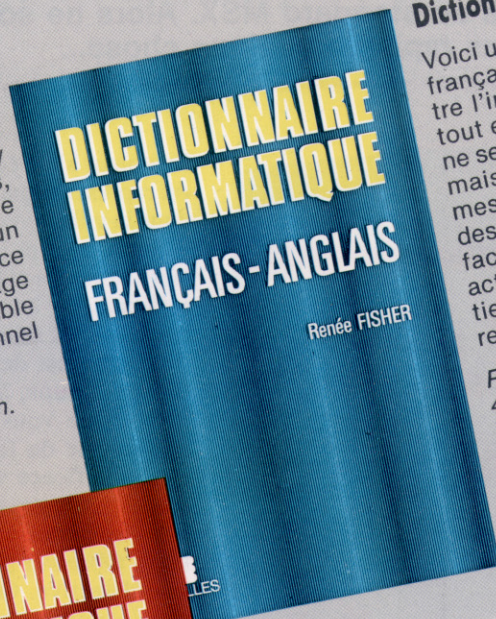
Par M. Ginguay et A. Lauret.  
328 pages.  
Masson.



**Dictionnaire micro-informatique et micro-électronique Cégos**

Définissant de façon claire plus de mille thèmes, y compris les plus récents, complété d'un lexique anglais-français et d'un lexique français-anglais, ce livre constitue un ouvrage de référence indispensable aussi bien au professionnel qu'à l'amateur.

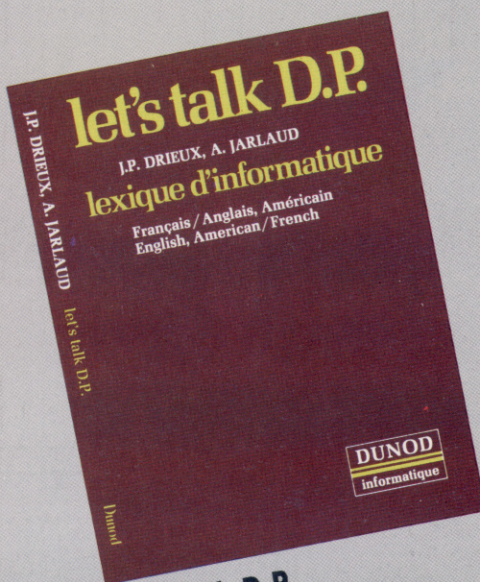
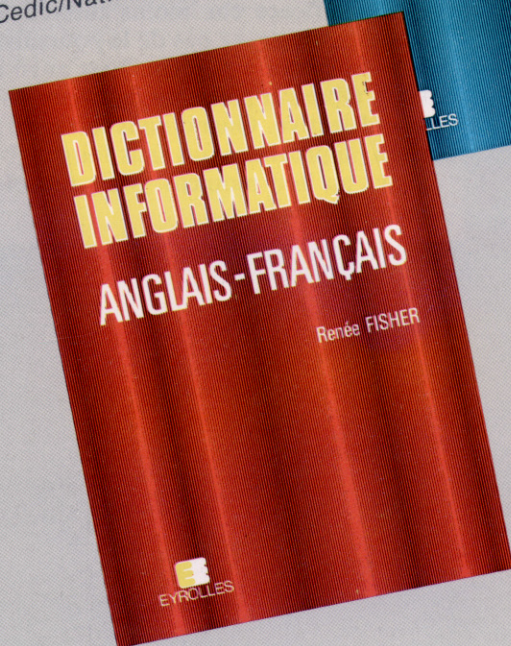
Par Christiane Morvan.  
Cedic/Nathan.



**Dictionnaire informatique**

Voici un dictionnaire anglais-français et français-anglais dont le but est de mettre l'informatique à la portée de tous, tout en se voulant complet. En effet, il ne se limite pas au vocabulaire courant, mais inclut un certain nombre de termes spécialisés, comme par exemple des termes juridiques ou commerciaux, facilitant grandement la lecture des actes de vente et des contrats de garantie, et tient compte des toutes dernières techniques, telle la télématique.

Par F. Fischer.  
480 pages.  
Troisième édition augmentée et mise à jour.  
Format 16 cm x 22 cm. 720 pages.  
Eyrolles.



**Dictionnaire micro-informatique**

Tout le vocabulaire de la micro-informatique familiale et professionnelle expliqué d'une façon simple et rigoureuse à travers 200 articles. Un index de plus de 600 mots vous permet d'accéder directement à l'article qui vous intéresse. Deux lexiques vous donnent les équivalents français-anglais et anglais-français de tous les mots clés de la micro-informatique.

Par Eric Duceau et Christophe Doe.  
Cedic/Nathan.

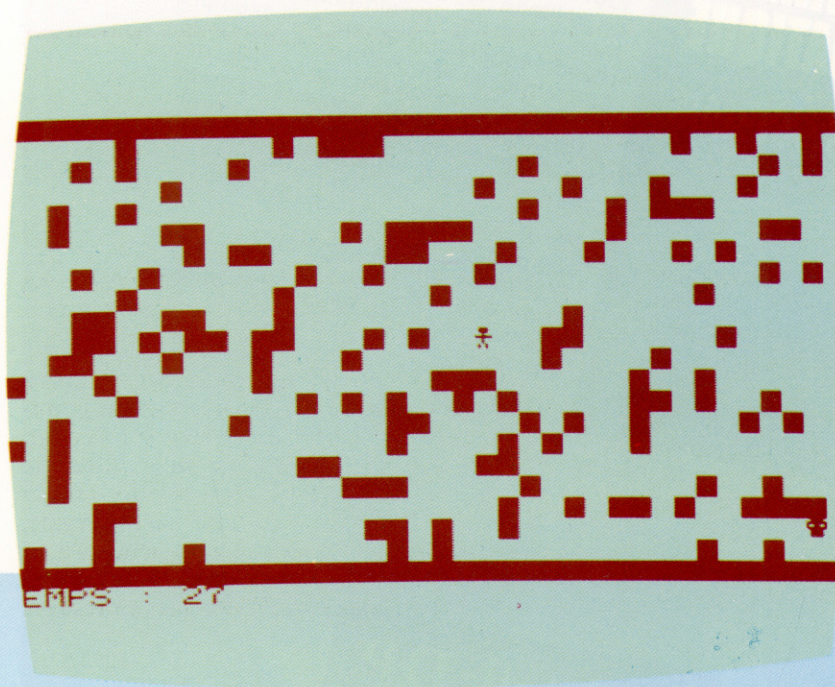
**Let's talk D.P. Lexique d'informatique français-anglais-américain english-american-french**

Les 3 000 mots anglais et français qui constituent le vocabulaire vivant et actuel de l'informatique sont ici regroupés par domaines définis, lorsque nécessaire, et présentés en situation dans des textes introductifs.

Par J.-P. Drieux et A. Jarlaud.  
146 pages.  
Dunod Informatique.

# Poursuite

Nous poursuivons notre présentation de programmes écrits pour les micros conçus au standard MSX. Alors ne soyez pas surpris si le titre de ce jeu vous rappelle quelque chose...



Le voleur s'est échappé, emportant le magot. (Il est représenté par un as de pique noir.) Il se cache dans la ville, et vous avez cinquante minutes pour le débusquer et l'arrêter. Attention, pas de précipitation! En effet, si vous vous jetez sur lui sans réfléchir, il y a toutes les chances pour qu'il vous file entre les doigts. La meilleure façon de vous y prendre est de l'aborder de côté. (Efficace à tous les coups, à condition de ne pas le rater!) Si vous ne vous sentez pas assez sûr de vous, attaquez-le de face, ce qui est plus facile mais moins efficace car moins discret. Encore un conseil : n'essayez pas de le poursuivre; cela ne vous mènerait à rien car il est aussi rapide que vous. Observez plutôt ses mouvements comme un vrai détective. Quand vous le verrez tourner en rond, approchez-vous sans faire de bruit et surgissez au bon moment. Mais rappelez-vous, le temps presse!

```

10 REM *****
20 REM * POURSUITE *
30 REM *****
40 KEY OFF
50 WIDTH 39
60 GOSUB 1060
70 S=0
80 N=32
90 V=128
100 P=129
110 GOSUB 670
120 ST=STICK(0)
130 DX=(ST=7)-(ST=3)
140 DY=(ST=1)-(ST=5)
150 Z=Z-.2
160 LOCATE 0,24,0
170 PRINT "TEMPS :";INT(Z+1);
180 IF Z<0 THEN 370
190 PX=PX+DX
200 PY=PY+DY
210 C=VPEEK(PX+PY*40)
220 IF C=V THEN 1010
230 IF C<>32 THEN PX=XP:PY=YP
240 VPOKE XP+YP*40,N
250 VPOKE PX+PY*40,P
260 YP=PY
270 XP=PX
280 VX=VX+CX
290 VY=VY+CY
300 IF VPEEK(VX+VY*40)<>N THEN GOSUB 510
310 IF VPEEK(VX+VY*40)<>N THEN 280
320 VPOKE XV+VY*40,N
330 VPOKE VX+VY*40,V
340 XV=VX
350 VY=VY
360 GOTO 120
370 IF INKEY#<>" " THEN 370
380 FOR I=1 TO 1000
390 NEXT I
400 LOCATE 10,6
410 PRINT "TEMPS ECOULE";
420 LOCATE 10,10
430 PRINT "SCORE :";S;
440 LOCATE 10,18,1
450 PRINT "UNE AUTRE ?";
460 D$=INKEY$
470 IF D$="" THEN 460
480 IF D$<>"N" AND D$<>"n" THEN RUN
490 CLS
500 END
510 DT=DT+1
520 GOSUB 620
530 IF VPEEK(XV+CX+(YV+CY)*40)=N THEN VX
=XV+CX:VY=VY+CY:RETURN
540 DT=DT-2
550 GOSUB 620
560 IF VPEEK(XV+CX+(YV+CY)*40)=N THEN VX
=XV+CX:VY=VY+CY:RETURN
570 DT=DT-1
580 GOSUB 620
590 VX=XV+CX
600 VY=VY+CY
610 RETURN
620 IF DT>4 THEN DT=DT-4
630 IF DT<1 THEN DT=DT+4
640 CX=(DT=1)-(DT=3)
650 CY=(DT=2)-(DT=4)
660 RETURN
670 CLS
680 FOR VX=0 TO 39
690 VPOKE VX+40,219
700 VPOKE VX+880,219
710 NEXT VX
720 FOR VY=2 TO 22
730 VPOKE VY*40,219
740 VPOKE VY*40+39,219
750 NEXT VY
760 FOR VX=1 TO 150
770 GOSUB 970
780 VPOKE PX+PY*40,219
790 NEXT VX
800 GOSUB 970
810 VX=PX
820 VY=PY
830 VPOKE VX+VY*40,V
840 XV=VX
850 VY=VY
860 GOSUB 970
870 VPOKE PX+PY*40,P
880 XP=PX
890 YP=PY
900 Z=50
910 CX=0
920 CY=0
930 DX=0
940 DY=0
950 DT=0
960 RETURN
970 PX=INT(RND(1)*37)+1
980 PY=INT(RND(1)*21)+2
990 IF VPEEK(PX+PY*40)<>N THEN 970
1000 RETURN
1010 FOR I=1 TO 5
1020 PLAY "DC"
1030 NEXT I
1040 S=S+1
1050 GOTO 110
1060 SCREEN 0,0
1070 COLOR 6,14
1080 CLS
1090 DEFINT A-Y
1100 PX=RND(-TIME)
1110 RESTORE
1120 FOR I=0 TO 15
1130 READ A
1140 VPOKE 3072+I,A*4
1150 NEXT I
1160 RETURN
1170 DATA 30,63,45,63,30,30,30,0
1180 DATA 28,28,8,62,8,8,20,20

```

**Page manquante  
(publicité)**

**Page manquante  
(publicité)**